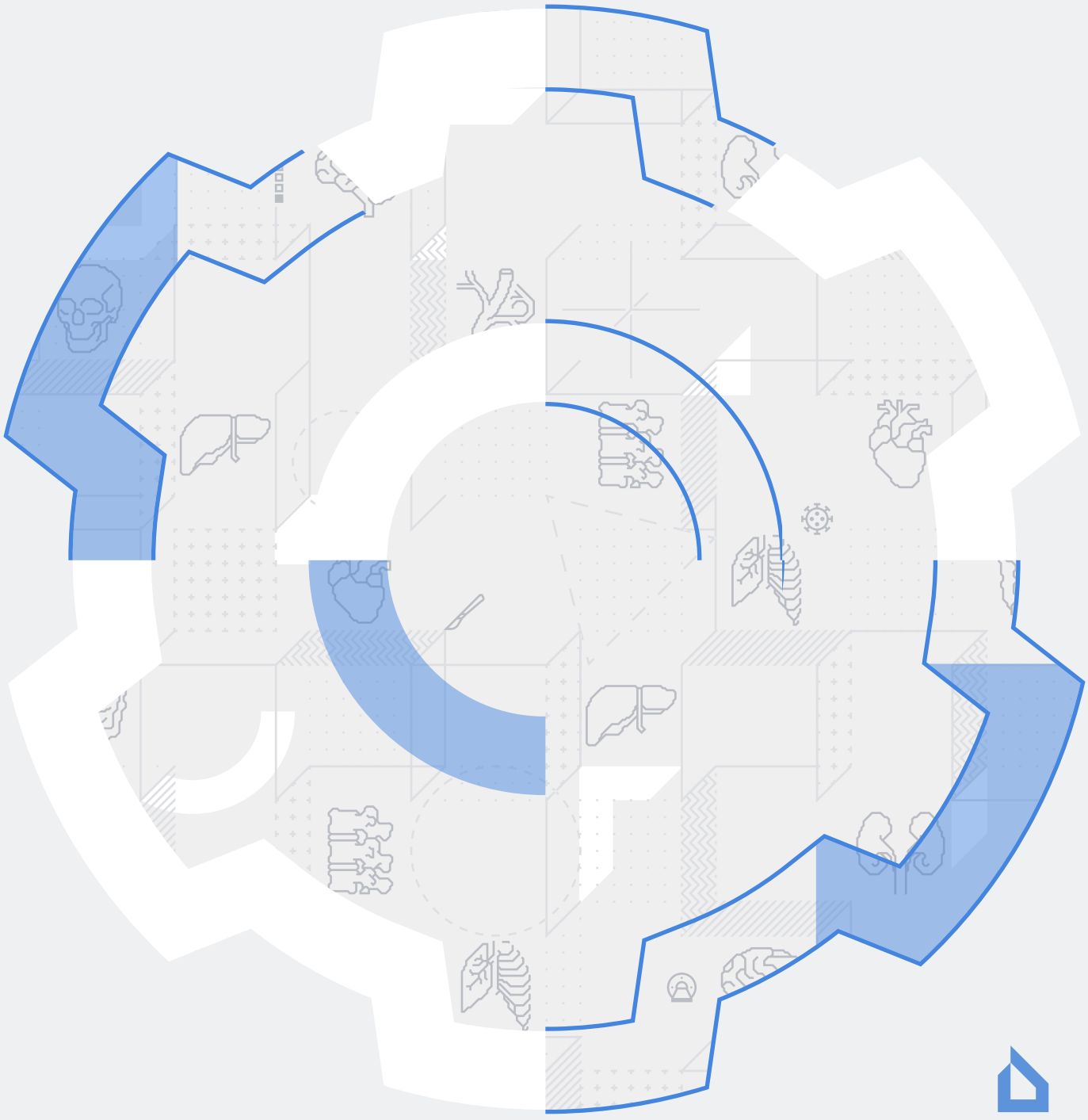
INOBITEC software for healthcare

Inobitec Web DICOM Viewer version 2.10

ADMIN'S MANUAL



The information contained in this User Manual belongs to Inobitec Software FZ-LLC, Dubai Media City, Building 05, Dubai, UAE, P.O. Box 73030. The Manual is delivered to the users of «Inobitec Web DICOM Viewer» software product exclusively for the purpose of working with this product. No part of the information contained herein can be modified, used for any other purpose or delivered to any third party without the prior written consent of Inobitec Software FZ-LLC. Inobitec Software FZ-LLC reserves the right to alter this Manual without prior notice.

Contents

About this Manual

The Admin's manual contains information on the system requirements and the structure of «Inobitec Web DICOM Viewer» package, as well as on its installation, deletion, launch, licensing, customization and integration with PACS servers.

The functions of the program, the role of the interface components and the tools are described in detail in the User's manual.

Accepted Conventions

Names of program interface elements, key names and important notes are printed in **bold**.
Image captions are printed in *italics*.

Technical Support

Technical support of «Inobitec Web DICOM Viewer» is provided by the Inobitec Software FZ-LLC specialists. When applying for technical support, please include the following information in your message:

- your computer OS name, version and bitness;
- he version of your operating system from which the client connects to the server;
- product version (for example, 2.10.2.317). The program version is specified in the **About** section of the administator's main menu. For details, see Section 3.13;
- type of the program build (unified or distributed) and the current DBMS version;
- product key and license key, if the license has been bought. To find the product key and the license key, select the **Server settings** option on the administrator's main menu. For details, see Section 3.2.

To apply for technical support, or if you have any further questions or comments, please email us at **support@inobitec.com**

Chapter 1

About the Product

The «Inobitec Web DICOM Viewer» is a client-server software. A web browser can be used as a client. Since all resource-intensive operations are performed on the server, the system requirements on which the client is installed are rather low. The «Inobitec Web DICOM Viewer» is intended for use in an infrastructure that uses one or more powerful servers and workstations whose resources are sufficient for the full functioning of modern web browsers.

64-bit versions are available for all the operating systems supported.

The «Inobitec Web DICOM Viewer» software product and installer does not:

- collect and transfer confidential user information;
- intercept network traffic;
- show ads;
- send spam;
- show messages not related to work;
- automatically update itself without notifying the user.

After uninstalling you do not need to restore your operating system and browser settings. Uninstalling is free of charge. Uninstalling does not adversely affect the operation of the computer and installed software. Files not related to the Web DICOM Viewer are not removed and changed after uninstalling.

Only specialists should use the «Inobitec Web DICOM Viewer» to make diagnoses.

To avoid personal data disclosure, secure channels should be used for data transfer via the network.

The Web DICOM Viewer functionality, installing, uninstalling, licensing are fully described in this User Manual on the website **inobitec.com**. License Agreement is available from the link **inobitec.com/eng/about/webviewerLic/**.

1.1 Destination

The «Inobitec Web DICOM Viewer» software product is intended for viewing, analyzing and printing medical data obtained from various DICOM equipment (modality). It is deployed on a server that can be accessed from client devices. The «Inobitec Web DICOM Viewer» software package can also be integrated with PACS servers.

The «Inobitec Web DICOM Viewer» is supplied as either a unified or a distributed build. The main features of the unified and distributed assemblies are outlined below.

1.2 Unified Build of the Web DICOM Viewer

The unified build is intended for use in scenarios with low load and a small number of users. As a database, either SQLite or PostgreSQL can be used. For the unified build of the Web DICOM Viewer, a trial period of 14 days is available with no functionality restrictions (see Section 3.3.1).

1.3 Distributed Build of the Web DICOM Viewer

The distributed build is designed for use in scenarios with high load and a large number of users. In this case, the resources of a single physical server may no longer be sufficient to handle the request flow. It is recommended to use PostgreSQL as the database.

The distributed build of the «Inobitec Web DICOM Viewer» program features a distributed architecture and consists of three nodes:

1. **Control Node** — This is the main node with which the user interacts. Only one copy of the control node is deployed. It provides static data (html-pages) to the user, ensures identification and authentication of the user, and is used as a gateway when performing operations with the storage and rendering servers.
2. **Storage Node** responsible for the storage of DICOM studies. It requires a database (PostgreSQL or other DBMS) and hard disk space to store user studies.
3. **Render Node** provides data visualization. The minimum system requirements for the server on which this node is deployed are indicated in the Section **System Requirements**.

Each of these nodes performs a specific function. Deployment of nodes can occur both on one server, and on different.

The software complex may comprise multiple storage nodes and render nodes. Nodes can be added as needed and as the load increases. To enable the use of resources from multiple physical machines simultaneously, the nodes must be deployed on separate physical machines (or on virtual machines located on different physical machines).

The following measures will ensure high efficiency and stable operation of the rendering server at high loads:

- make sure that the swap file is big enough to support stable operation during peak loads;
- provide for sufficient amount of RAM in order to minimize the number of occasions when the swap file is accessed, as well as to ensure rapid processing of rendering tasks;
- do not use the **--no-cache** parameter for **rendernode** so as to engage caching and speed up the recurring rendering operations. This will help you reduce the load on the CPU and the disk subsystem;
- use the distributed architecture of the rendering system with a sufficient number of nodes.

The distribution of loads between rendering nodes is based on the number of active sessions. When a new session is created, it is assigned a rendering node from the available nodes with the fewest assigned sessions. When operations are performed within the current session, all rendering requests will be sent to this selected node.

The distribution of loads between storage nodes is achieved by assigning a storage node to each user. The node is assigned by the administrator when the user is created. All of the user's storage data are located on this node. The information about the user's personal PACS servers and their SCU is also stored on this node. All the operations with PACS servers during this user's sessions are performed using this node. For sessions with an unknown user (e.g., opening a study via a link from a global PACS server), the storage node is selected in compliance with the rendering node selection logic — the node with the fewest sessions is chosen.

1.4 System Requirements

Minimum System Requirements for Server:

Operating system:

- Windows version not lower than Windows 7 x64;
- Ubuntu 20 x64;

DBMS: PostgreSQL 9.6 and greater (optional, depends on the selected DBMS);
processor: Intel Core i3-2100 or similar:

- clock frequency of 3,1 GHz;
- number of processor cores 2;
- maximum number of threads 4;

RAM: 16 GB;
If the server is deployed in a virtual machine, it is recommended to allocate static RAM, not dynamic RAM.
video card: GeForce GT 710 or similar:

- clock frequency of 954 MHz;

- the amount of video memory 2 GB;

free space on the disk: 100 MB (without space for the log and the storage);
network card.

Recommended System Requirements for Server:

Operating system:

- Windows version not lower than Windows 7 x64;
- Ubuntu 20 x64;

DBMS: PostgreSQL 9.6 and greater (optional, depends on the selected DBMS);
processor: Intel Core i7-6700K or similar:

- clock frequency of 4,0 GHz;
- number of processor cores 4;
- maximum number of threads 8;

RAM: 32 GB;
If the server is deployed in a virtual machine, it is recommended to allocate static RAM, not dynamic RAM.
video card: GeForce GTX 1080 or similar;

- clock frequency of 1,6 GHz;
- the amount of video memory 8 GB;

free space on the disk: 100 MB (without space for the log and the storage);
network card.

Minimum System Requirements for Client:

processor:

- clock frequency of 1,5 GHz;
- number of processor cores 1;

RAM: 2 GB;
video card: without hardware acceleration support;
network card;
display;
keyboard: standard;
mouse: two-button with a scroll wheel;
Internet browser.

Chapter 2

Installing, Uninstalling and Launching the Program

2.1 Installation of the Unified Build of the Program on Windows OS

For the unified build of the Web DICOM Viewer, SQLite or PostgreSQL can be used as the DBMS.

If SQLite is used, no prior installation of the DBMS is required.

If PostgreSQL is intended to be used as the DBMS, make sure that PostgreSQL is installed on the system before installing the Web DICOM Viewer. The Web DICOM Viewer software package utilizes PostgreSQL version 9.6 or higher as the DBMS for its operation.

2.1.1 Installing the Unified Build with the Installer

To install the program, proceed as follows:

1. Run the file by double-clicking on it.

2. Click **CONTINUE** in the window shown in Fig. 2.1.

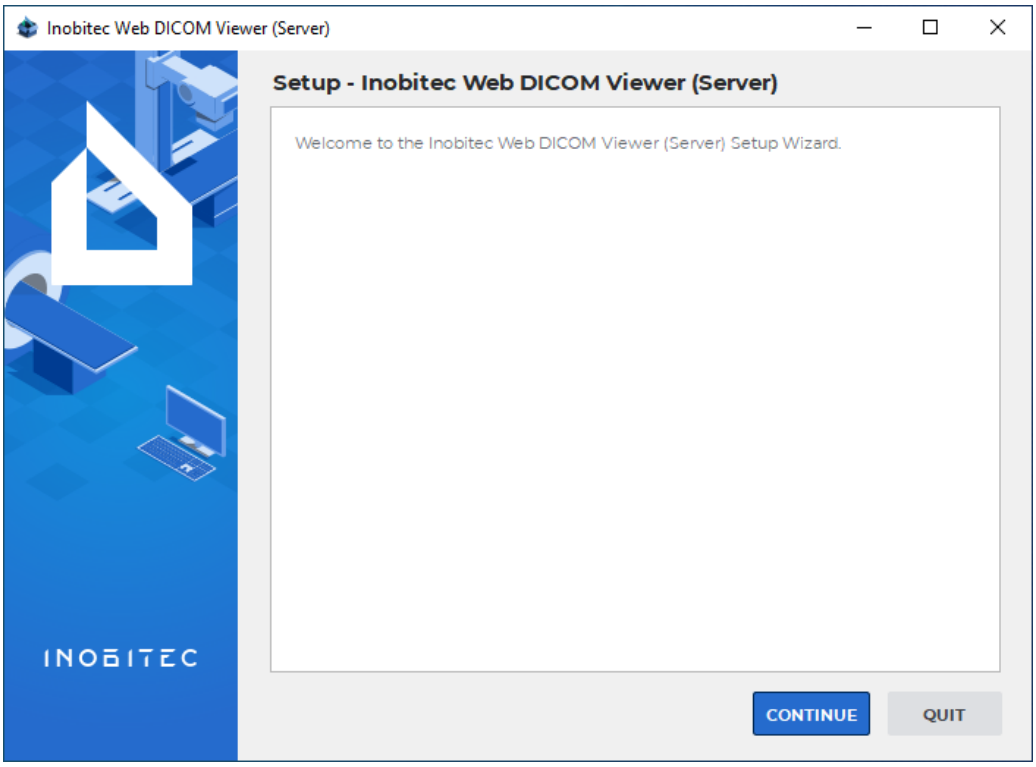


Figure 2.1: Beginning of the installation

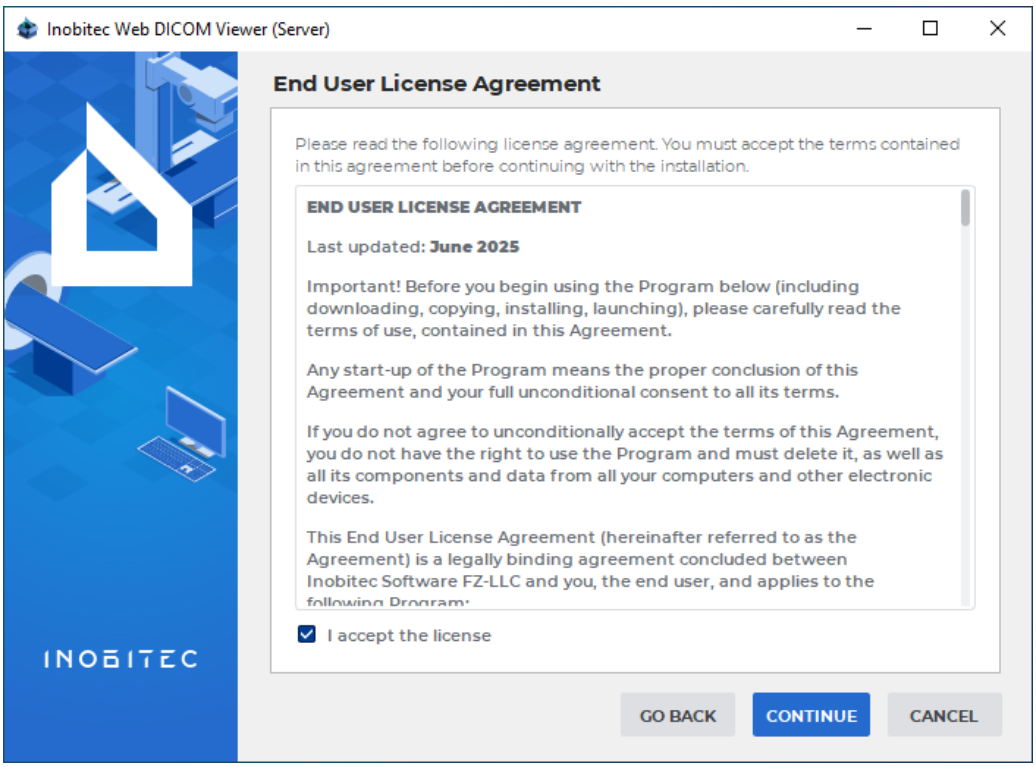


Figure 2.2: License Agreement

3. Read the License Agreement carefully. Its text is also available at **inobitec.com/eng/about/webviewerLic/**. If you accept the terms of the License Agreement, check the **I accept the license** box and click **CONTINUE** (see Fig. 2.2). Otherwise, click **CANCEL** and choose **YES** in the confirmation dialog box. To get back to the previous step, click **GO BACK**.

4. Select the users for whom the program is being installed (Fig. 2.3). The current user option is selected by default. The program is installed in the user's home directory.

   If you select the **Anyone who uses this computer (all users)** option, the program will be installed in the system directory. To do this, you need administrator rights. To install the program as a service, check the **Install program as a service** box (the option is selected by default). Select one of the two options for running the program: **Automatically** (when the operating system is launched) or **Manually**. The automatic option is selected by default.

   To get back to the previous step, click **GO BACK**. To continue the installation, click **CONTINUE**.

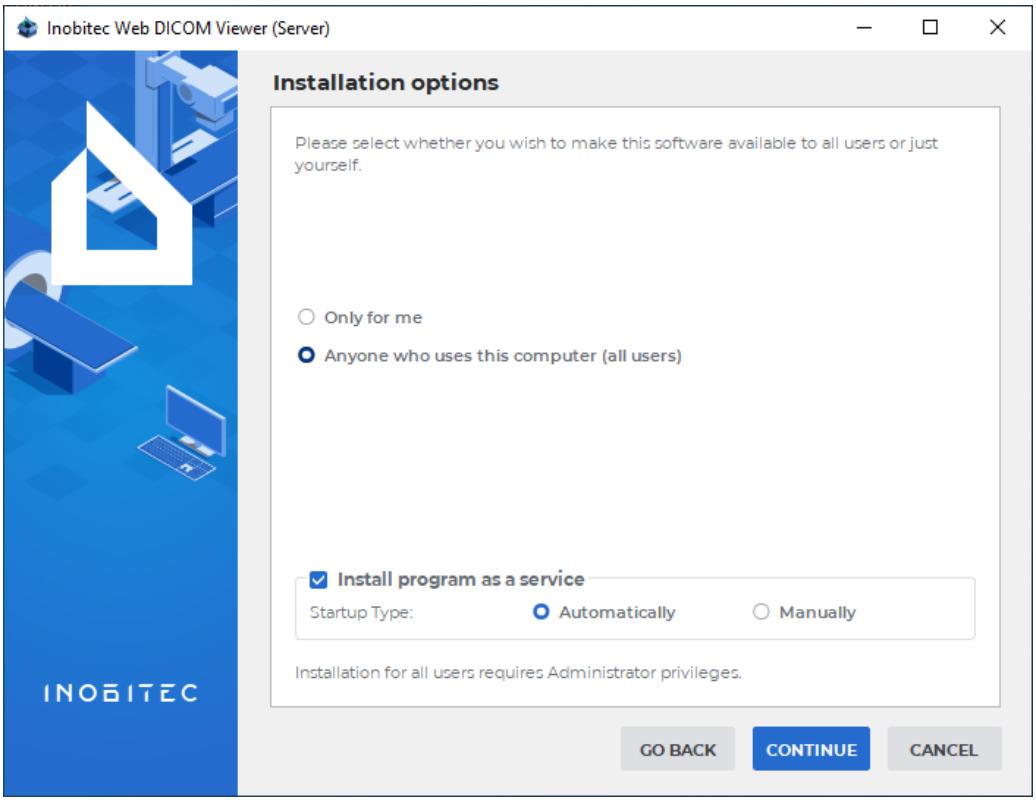


Figure 2.3: Select users

5. If required, change the installation folder (Fig. 2.4). We recommend using the default value. To get back to the previous step, click **GO BACK**. To continue the installation, click **CONTINUE**.

Attention!

- **do not provide paths containing only the root disk (e.g. D:\). Otherwise, all the data will be deleted from that disk when the program is deleted;**
- **do not specify directories created by other users or third-party software. Otherwise, you may lose data when the directories are deleted.**

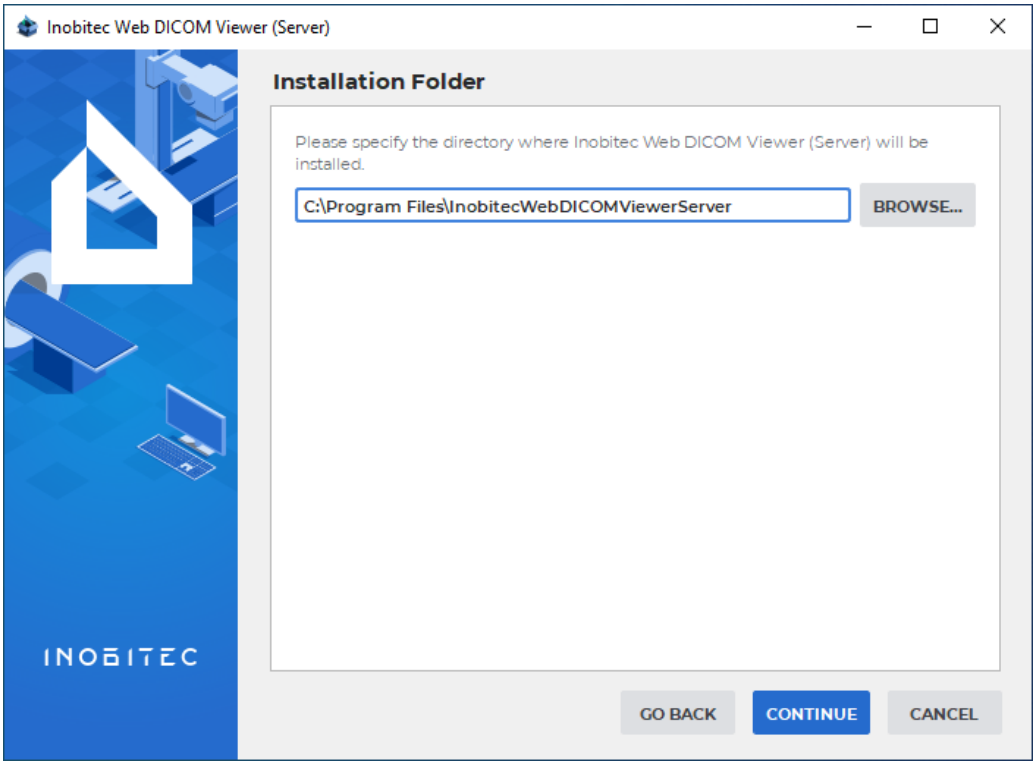


Figure 2.4: Selecting the installation folder

6. If the program has already been installed in the selected installation folder, the following warning will pop up: **The directory you selected already exists and contains an installation. To install the product the installation will be uninstalled first.** To uninstall the program, click **YES**. The uninstaller will be activated in a new window. The process may take some time. Click **DONE** to go to the next installation step.

   To keep the installed program version and select a new installation folder, click **NO** in the window shown in Fig. 2.4 and specify a different directory.

7. In the **Logging** window (Fig. 2.5), specify the directory where the log files are to be stored and click **CONTINUE**. We recommend using the default value.

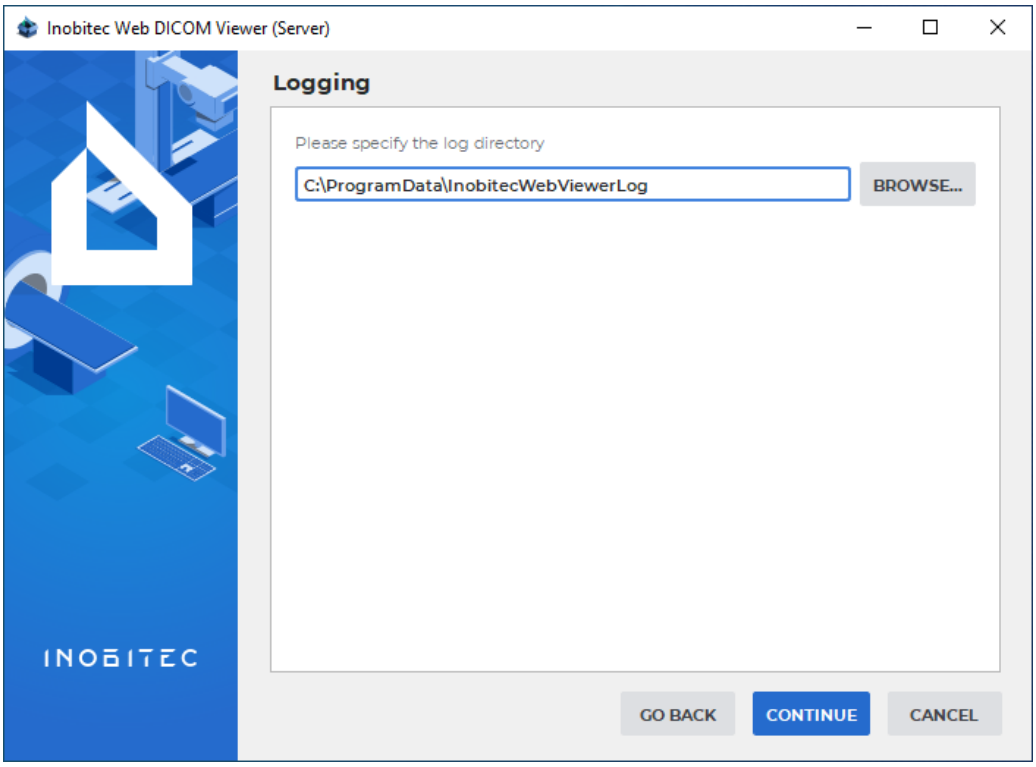


Figure 2.5: The **Logging** *window*

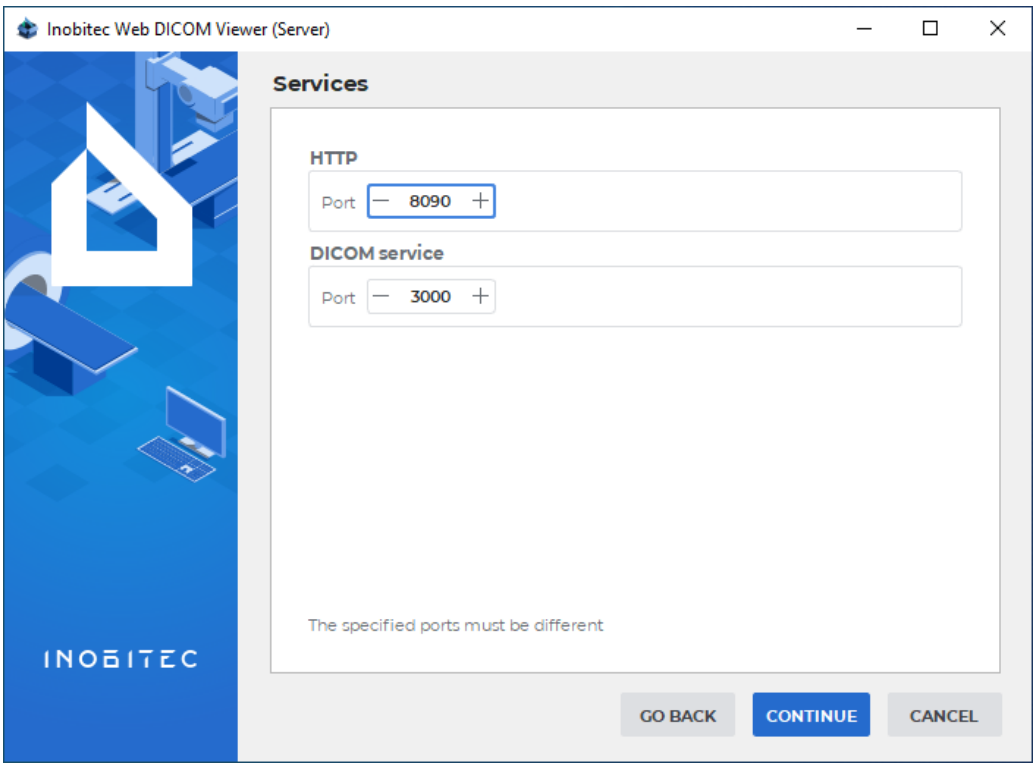


Figure 2.6: The **Services** *window*

8. In the **Services** window (Fig. 2.6), specify the RenderServer port value in the **HTTP** field (the default value is 8090), and in the **DICOM service** field, specify the value of the DICOM Service port for file acceptance (the default value is 3000). When a value is entered, the port's availability is checked automatically. If the port is engaged, a warning pops up on the right-hand side and the **CONTINUE** button becomes inactive.

9. If required, change the directory on the **Start** menu to create shortcuts for the program (Fig. 2.7). We recommend using the default value. Click **CONTINUE**.

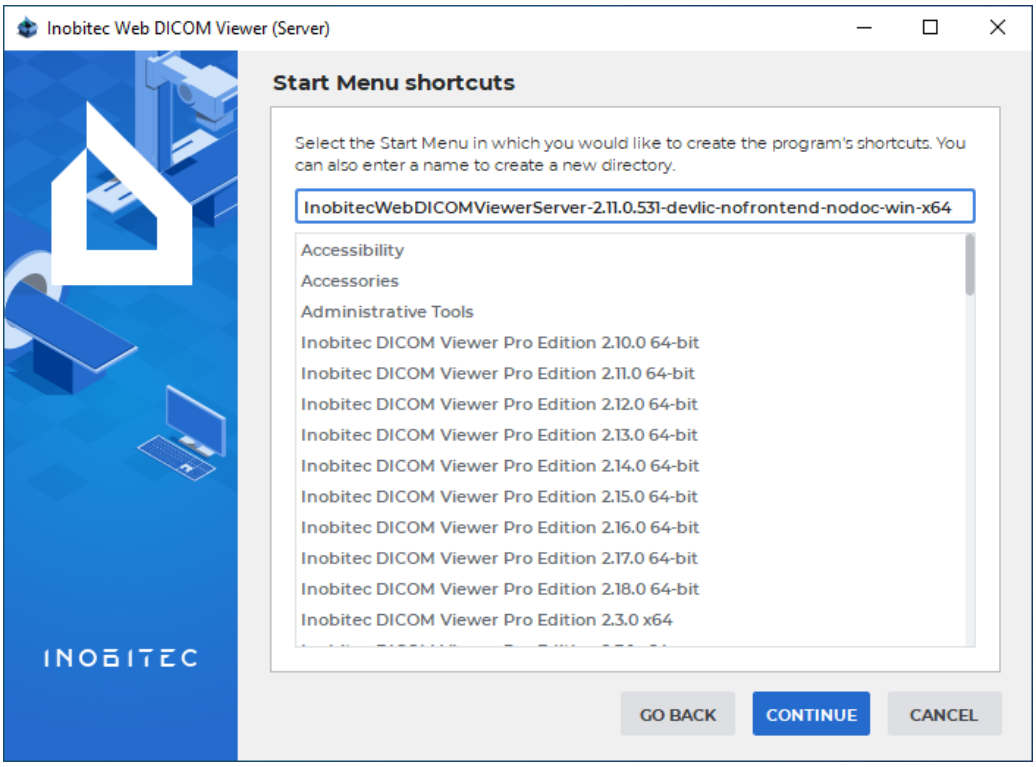


Figure 2.7: Directory for shortcuts creation on the **Start** *menu*

10. Click **INSTALL** (Fig. 2.8) in the Installer window to confirm the program installation. Wait until the installation is completed.

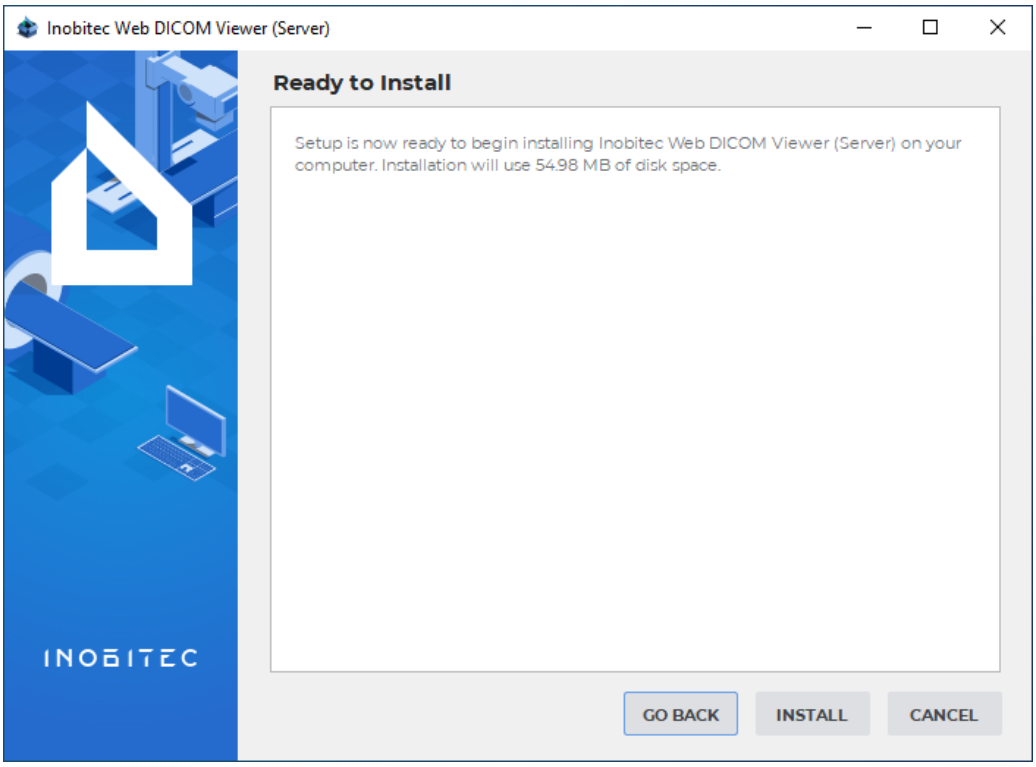


Figure 2.8: Installation confirmation

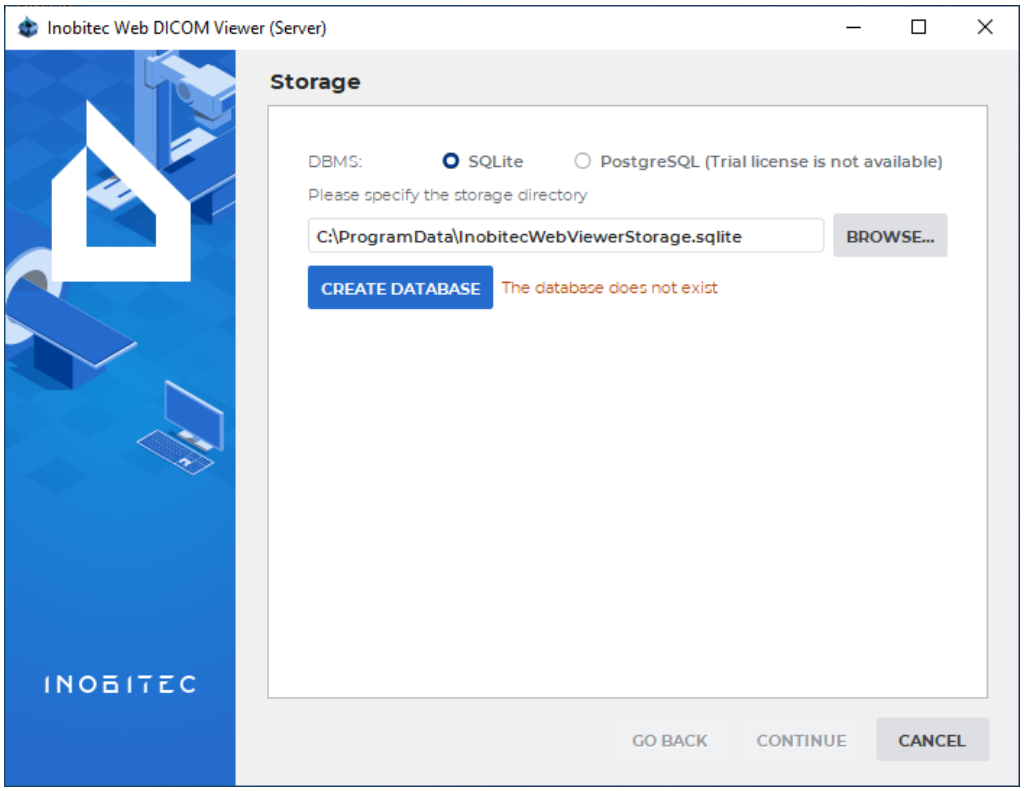


Figure 2.9: The **Storage** *window*

11. In the **Storage** window (Fig. 2.9), select the DBMS to be used for work with the program:

    - **SQLite** (selected by default). SQLite database customization and deployment is described in Section 2.1.2;
    - **PostgreSQL**. PostgreSQL database customization and deployment is described in Section 2.1.3.

2.1.2 SQLite Database Customization and Deployment on the Unified Program Build

1. In the **Storage** window (Fig. 2.10), set the DBMS switch to **SQLite** (the default setting).
2. Edit the path to the storage directory and the SQLite database file. The parameter may be used by default. The default path to the storage directory is:

    - *<USERPROFILE>\InobitecWebViewerStorage.sqlite*, if installation for the current user has been selected;
    - *<ProgramData>\InobitecWebViewerStorage.sqlite*, if installation for all users has been selected.

    Attention! The SQLite database file is named webviewer-database and placed in the storage directory.

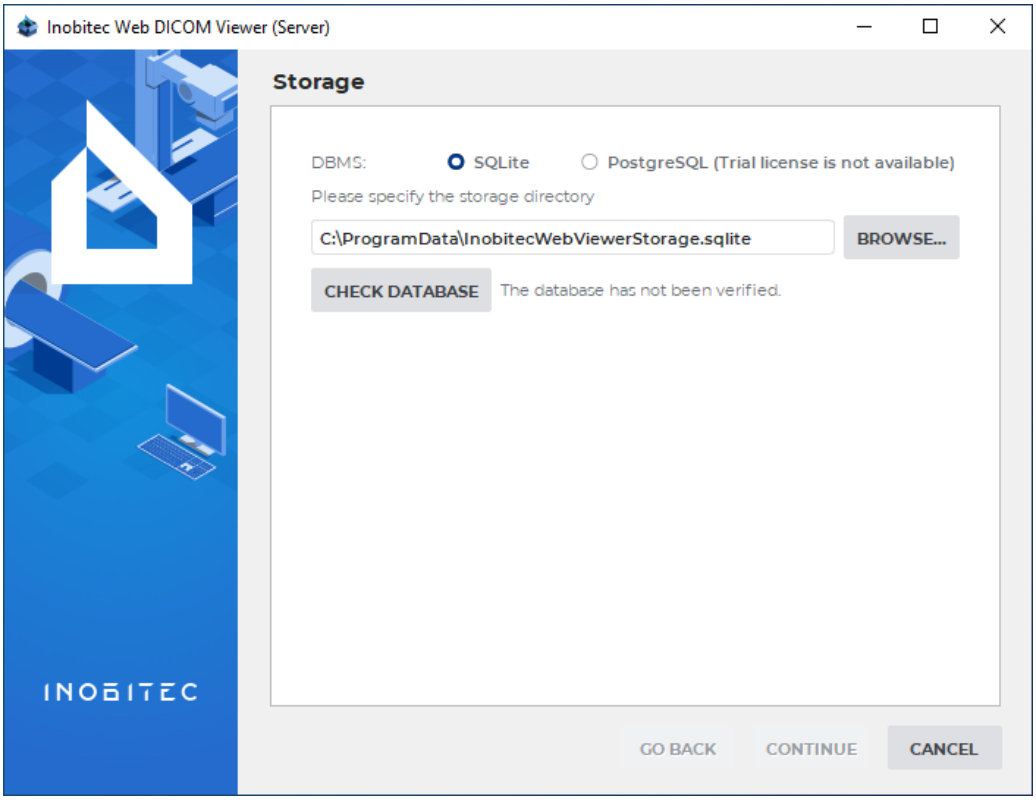


Figure 2.10: Checking up on the SQLite database

3. If the path to the storage directory has been edited or if the DBMS switch position has been changed, check the database state by clicking the **CHECK DATABASE** button. When you click the **CHECK DATABASE** button, the specified parameter values are saved in the **renderserverconfig.json** and **renderserverconfig-sqlite.json** configuration files. After the parameters have been successfully saved, the database state is checked.

   The possible database states are:

   - **The database is up to date**. The database is updated and ready to use;
   - **Internal error**. The error may be due to invalid database configuration parameters or a problem with the DBMS driver initialization;
   - **The database does not exist**. The **CREATE DATABASE** button is displayed;
   - **The database is empty**. The database exists, but does not contain any tables. The **CREATE DATABASE TABLES** button is displayed;
   - **Invalid data of database**. The database contains tables, but the relevant tables are missing or the existing tables do not contain the relevant fields. You need to choose a different database. To do this, change the path to the storage directory and check the database state;
   - **The content of database is outdated**. The contents of some tables may be outdated. You need to choose a different database. To do this, change the path to the storage directory and check the database state;
   - **Unsupported database version**. The database corresponds to the new program version, which is incompatible with the current server version. You need to choose a different database. To do this, change the path to the storage directory and check the database state;
   - **The database version is outdated**. Upgrade the database to the new version. The **UPDATE DATABASE** button will be displayed.

4. To create a new database, click the **CREATE DATABASE** button. The process may take some time.

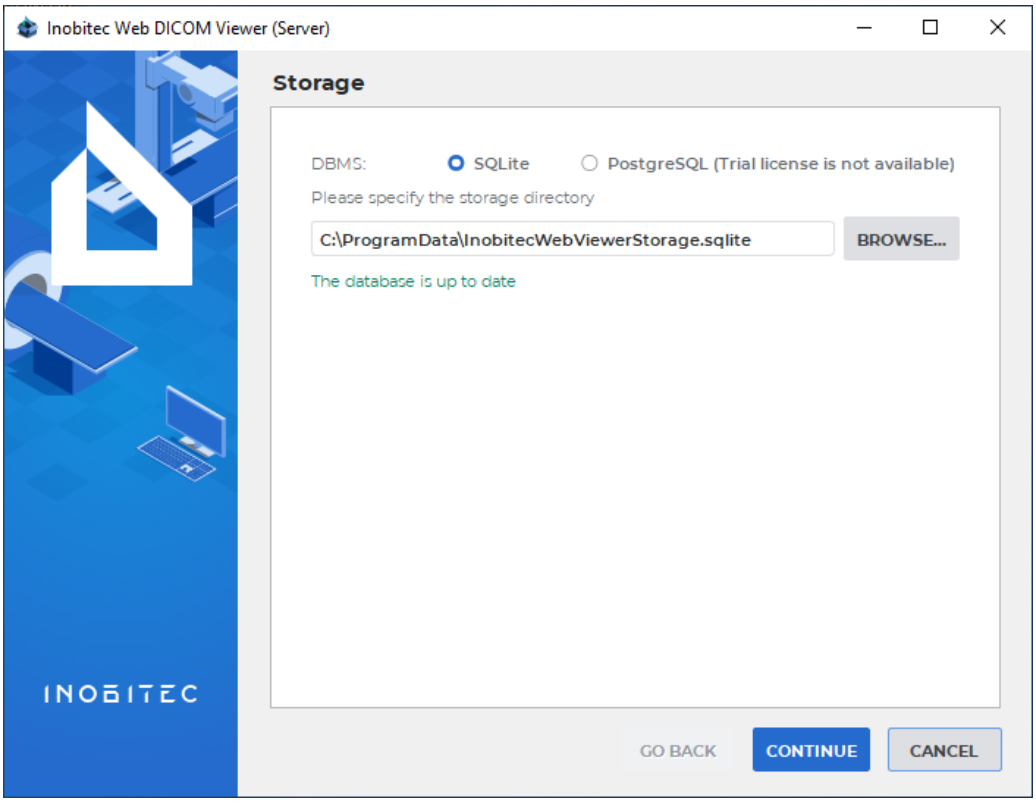


Figure 2.11: The database creation process has been completed

5. To go to the **License Activation** window, click the **CONTINUE** button (Fig. 2.11). The license activation procedure is described in Section 2.1.4.

2.1.3 PostgreSQL Database Customization and Deployment on the Unified Program Build

Before deployment and customization of the database scheme, make sure that PostgreSQL DBMS version 9.6 or superior has been installed on the server and that the administrator has been assigned a login and a password.

Make sure that the path to the **bin** folder of the PostgreSQL DBMS has been added to the **PATH** environment variable.

Before database customization, make sure that PostgreSQL service has been activated.

1. In the **Storage** window (Fig. 2.12) , set the DBMS switch to **PostgreSQL**.

2. Edit the path to the storage directory. The parameter may be used by default. The default path to the storage directory is:

   - *<USERPROFILE>\InobitecWebViewerStorage.postgres*, if installation for the current user has been selected;
   - *<ProgramData>\InobitecWebViewerStorage.postgres*, if installation for all users has been selected.

3. In the **Database options** section, edit the following parameters (if required):

   - **Database name** (the default value is **webviewer**);
   - **Host** — the address of the database server (the default value is **localhost**);
   - **Port** — the port of the database server (the default value is **5432**);
   - **User name** for the database (the default value is **webviewer**);
   - **Password** for the database user (the default value is **webviewer**).

   Attention! If automatic initialization of the PostgreSQL DBMS driver with the help of the installer has failed, the Storage window will only display the field for entering the path to the directory with the dynamic library of the PostgreSQL DBMS driver (libpq.dll), as well as the CHECK DATABASE button.

   Enter the path to the directory with the library of the PostgreSQL DBMS driver and check the database state.

4. Check the database state by clicking the **CHECK DATABASE** button. When you click the **CHECK DATABASE** button, the specified parameter values are saved in the **renderserverconfig.json** and **renderserverconfig-postgresql.json** configuration files. After the parameters have been successfully saved, the database state is checked.

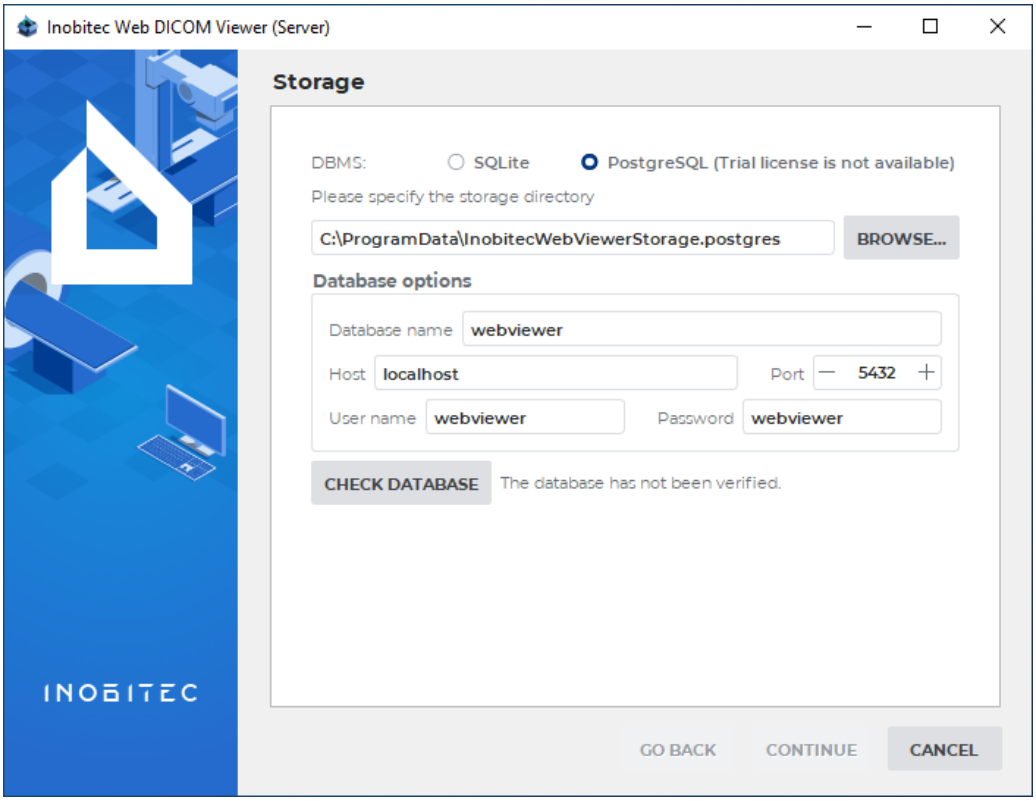


Figure 2.12: Checking up on the PostgreSQL database

The possible database states are:

- **The database is up to date**. The database is updated and ready to use;

- **Internal error**. The error may be due to invalid database configuration parameters or a problem with the DBMS driver initialization;
- **The database does not exist**. The **CREATE DATABASE** button is displayed;
- **The database is empty**. The database exists, but does not contain any tables. The **CREATE DATABASE TABLES** button is displayed;
- **Invalid data of database**. The database contains tables, but the relevant tables are missing or the existing tables do not contain the relevant fields. You need to choose a different database. To do this, change one of the following parameters: the name of the database, the host or the port;
- **The content of database is outdated**. The contents of some tables may be outdated. You need to choose a different database. To do this, change one of the following parameters: the name of the database, the host or the port;
- **Unsupported database version**. The database corresponds to the new program version which is incompatible with the current server version. You need to choose a different database. To do this, change one of the following parameters: the name of the database, the host or the port;
- **The database version is outdated**. Upgrade the database to the new version. The **UPDATE DATABASE** button will be displayed.

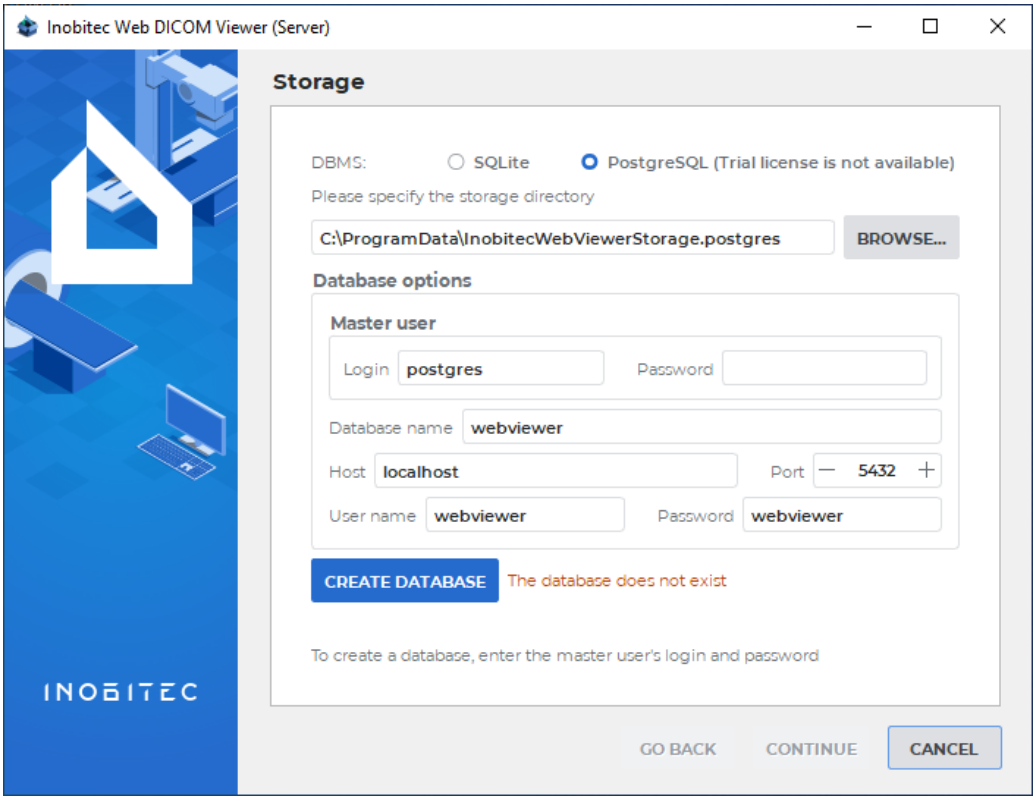


Figure 2.13: Entering the DBMS administrator login and password

5. To create a new database, click the **CREATE DATABASE** button. In PostgreSQL, you can only create a database as the DBMS administrator, so in the **Database options** section (see Fig. 2.13), you need to edit the PostgreSQL DBMS administrator parameters:

- **Login** for the DBMS administrator (the default value is **postgres**);
- **Password** for the DBMS administrator (no default value is provided).

Click the **CREATE DATABASE** button again. The database creation process may take some time.

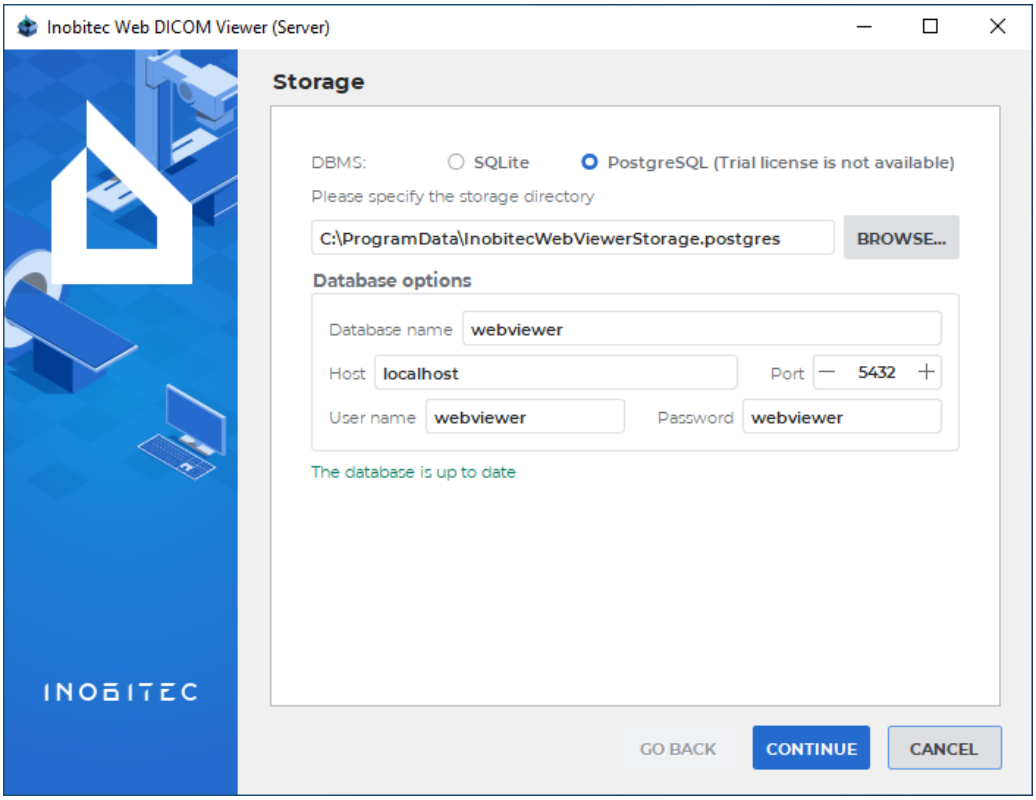


Figure 2.14: The database creation process has been completed

6. To go to the **License Activation** window, click the **CONTINUE** button (see Fig. 2.14). The license activation procedure is described in Section 2.1.4.

2.1.4 License Activation

In the **License Activation** window (Fig. 2.15), the current license is checked and the trial or the single license is activated. In the window, you can see the product code and the state of the current license.

1. To activate a license with a license key, enter the license key in the respective field and click the **ACTIVATE** button.
2. To test the Web DICOM Viewer for a trial period, click the **START TRIAL PERIOD** button. The trial period is only available for the unified program with SQLite DBMS. For details on the trial period, see Section 3.3.1.

   Attention! Self-activation of the trial period through the program interface is not available if the Web DICOM Viewer is:

- **configured to use PostgreSQL DBMS;**
- **installed in a virtual environment.**

In this case, the **START TRIAL PERIOD** button is not displayed in the **License Activation** window. To obtain a trial period for the program that uses PostgreSQL DBMS or is installed in a virtual environment, you must contact the support service at **support@inobitec.com**.

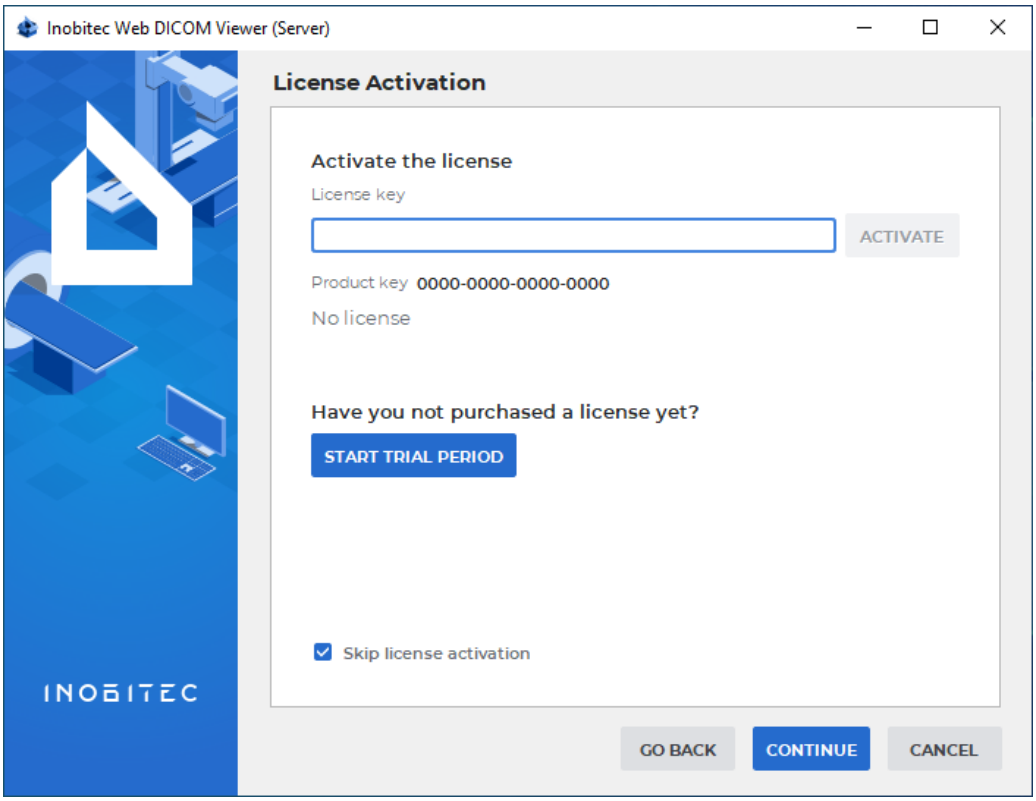


Figure 2.15: License activation

3. To proceed with installation without license activation, check the **Skip license activation** box and click the **CONTINUE** button. After the program has been installed, the license may be activated on the **Server settings** page on the Admin Panel (see Sections 3.2 and 3.3).

4. In the **Completing the installation of Inobitec Web DICOM Viewer (Server)** window (Fig. 2.16), check the actions to be performed after the installation has been completed:

   - **Run Inobitec Web DICOM Viewer Server**. Upon completing the installation, the server will be activated either in a separate window or as a service, if the **Install program as a service** box has been checked (Fig. 2.3). If the box is unchecked, the other options are hidden;
   - **Open Admin Panel**. Upon completing the installation, a box will appear where you will have to enter the credentials in order to go to the Admin Panel;

- **Open Inobitec Web DICOM Viewer**. Upon completing the installation, a box will appear where you will have to enter the user's credentials in order to enter the program.

5. To complete the installation procedure, click **DONE**.

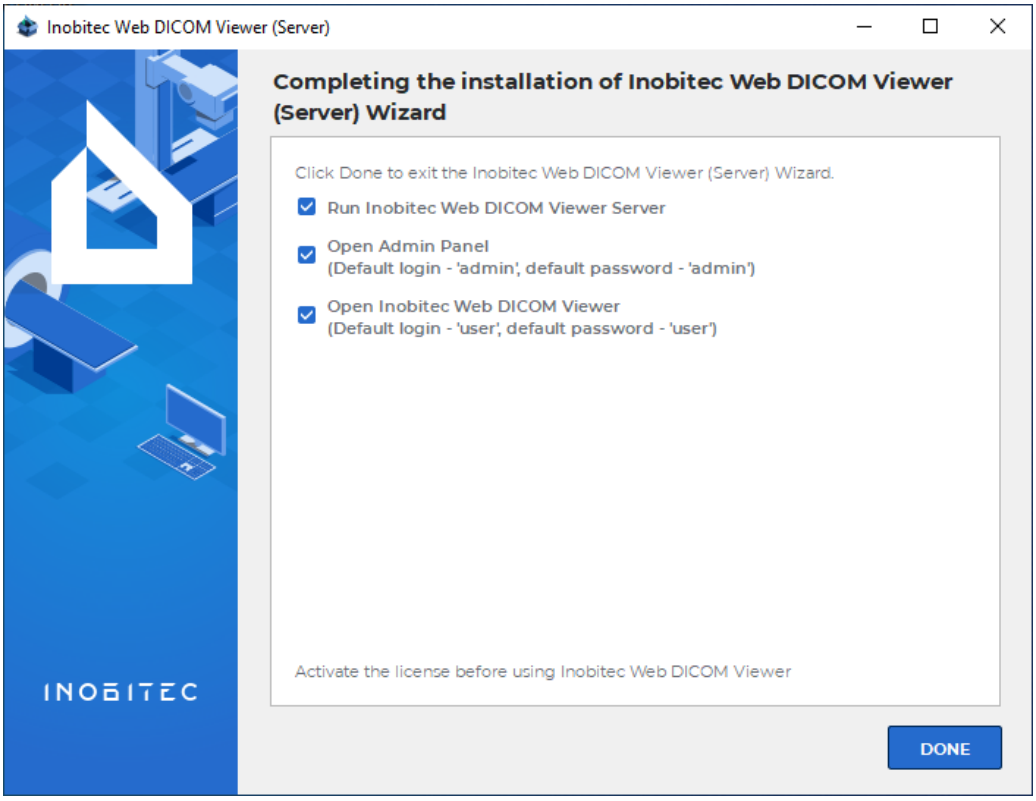


Figure 2.16: Completing the Installation

2.2 Installation of the Distributed Build of the Program on Windows OS

For the distributed build of the Web DICOM Viewer, it is recommended to use PostgreSQL as the DBMS.

2.2.1 Preparation for Installation

1. Install the *Runtime* redistributable component library for Microsoft Visual C++ (abbreviated as MSVC) 2019 on the PC using the respective *vc_redist.x64.exe* package.
2. If PostgreSQL is used as the DBMS, install PostgreSQL on the system and assign a login and password to the DBMS administrator.
3. Ensure that the path to the **bin** folder of the PostgreSQL DBMS is added to the **PATH** system environment variable.

2.2.2 Installation and Launch of the Control Node

1. Extract the distribution files from the archive into a separate directory.
2. Edit the *control.json* file modifying the node port and other parameters (optional). The parameter values in the configuration file can be left unchanged.
3. Edit the *controlnodeconfig-postgresql.json* file specifying the values for the following database parameters:
   - **driver** — database driver name (default value: *QPSQL*);
   - **name** — node database name (default value: *control*);
   - **host** — database server address (default value: *localhost*);
   - **port** — database server port (default value: *5432*);
   - **user** — database user name (default value: *control*);
   - **password** — database user password (default value: *control*).
4. To create a database for the control node of the Web DICOM Viewer, run the *ControlNode.exe* executable file from the command line with the following parameters:

   ControlNode.exe -c controlnodeconfig-postgresql.json --create-db-user --create-database --master-db-user <master_user_name> --master-db-password <master_db_password>,

   where

   <master_user_name> — is the name of a database user with administrator rights to create new users and databases (default: *postgres*);

   <master_db_password> — is the password of the database user.

   If the database user specified in step 3 already exists, the *--create-db-user* flag is not required.
5. To launch the control node, run the executable file *ControlNode.exe* with the command-line parameter *-c controlnodeconfig-postgresql.json*.

   ControlNode.exe -c controlnodeconfig-postgresql.json

   Alternatively, simply rename the file *controlnodeconfig-postgresql.json* to *controlnode-config.json* beforehand, in which case specifying it as a command-line parameter is not necessary.

2.2.3 Installation and Launch of the Storage Node

1. Extract the distribution files from the archive into a separate directory.
2. Edit the *storage.json* file modifying the node port and other parameters (optional). The parameter values in the configuration file can be left unchanged.
3. Enter the address of the control node (ControlNode) in the **control** parameter of the *storage.json* file.
4. Edit the *storagenodeconfig-postgresql.json* file specifying the values for the following database parameters:

- **storage** — path to the storage directory (default value: *~/InobitecWebViewerStorage.postgres*);
- **driver** — database driver name (default value: *QPSQL*);
- **name** — node database name (default value: *storage*);
- **host** — database server address (default value: *localhost*);
- **port** — database server port (default value: *5432*);
- **user** — database user name (default value: *storage*);
- **password** — database user password (default value: *storage*).

5. To create the database for the storage node of the Web DICOM Viewer, run the *StorageNode.exe* executable file from the command line with the following parameters:

   StorageNode.exe -c storagenodeconfig-postgresql.json --create-db-user --create-database --master-db-user <master_user_name> --master-db-password <master_db_password>,

   where

   <master_user_name> — is the name of a database user with administrator rights to create new users and databases (default: *postgres*);

   <master_db_password> — is the password of the database user.

   If the database user specified in step 4 already exists, the *--create-db-user* flag is not required.

6. To launch the storage node, run the *StorageNode.exe* executable file with the following command-line parameter *-c storagenodeconfig-postgresql.json*.

   StorageNode.exe -c storagenodeconfig-postgresql.json

   Alternatively, simply change the name of the file from *storagenodeconfig-postgresql.json* to *storagenodeconfig.json* beforehand, in which case specifying it as a command-line parameter is not necessary.

2.2.4 Installation and Launch of the Rendering Node

1. Extract the distribution files from the archive into a separate directory.

2. Run the *RenderNode.exe* executable file.

2.2.5 Registration and Launch of the Server as a Service

1. Launch the command prompt as an administrator. Specify the path to the directory containing the unzipped distribution files by running the following in the command prompt:

   set INSTALL_DIR=<path to the installation directory>
   set OPTIONS=%OPTIONS% --no-log-to-console

2. Register the nodes as services by running the following in the command prompt:

for the ControlNode:
sc create "Inobitec Web Viewer Control Node" binPath= "%INSTALL_DIR%\ControlNode.exe %OPTIONS% -c %INSTALL_DIR%\controlnodeconfig.json --log %INSTALL_DIR%\controlnode.log"
for the StorageNode:
sc create "Inobitec Web Viewer Storage Node" binPath= "%INSTALL_DIR%\StorageNode.exe %OPTIONS% -c %INSTALL_DIR%\storagenodeconfig.json --log %INSTALL_DIR%\storagenode.log"
for the RenderNode:
sc create "Inobitec Web Viewer Render Node" binPath= "%INSTALL_DIR%\RenderNode.exe %OPTIONS% --log %INSTALL_DIR%\rendernode.log"

The created services are launched in the standard manner.

2.3 Uninstalling the Program on Windows OS

2.3.1 Uninstalling the Unified Program Build with the Uninstaller

To uninstall the Web DICOM Viewer, proceed as follows:

1. Go to the directory where the program has been installed and run the **maintenance-tool.exe** uninstaller.

2. In the window shown in Fig. 2.17, select the **Remove all components** option and click **CONTINUE**.

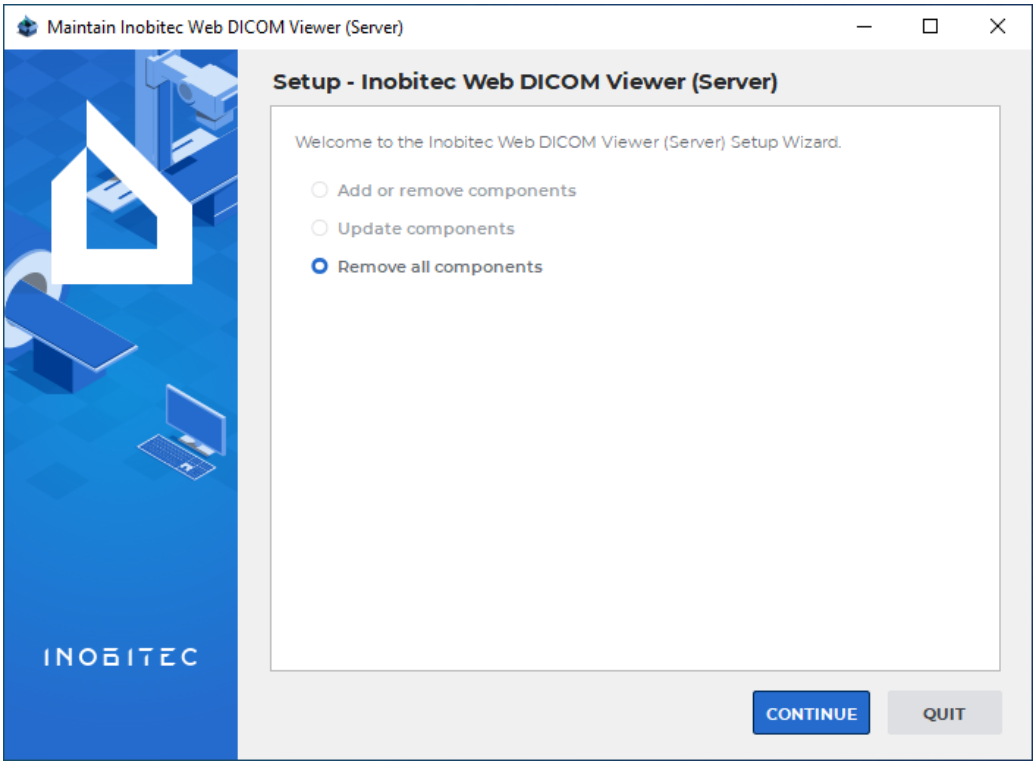


Figure 2.17: Uninstalling the program

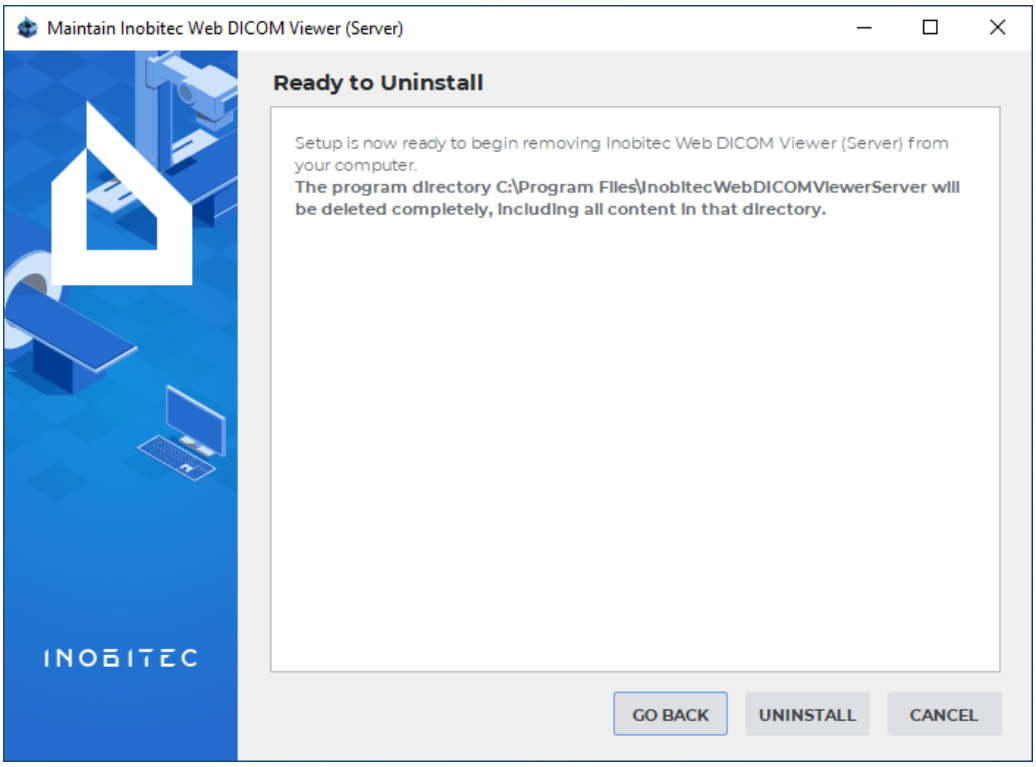


Figure 2.18: Confirming the program deinstallation

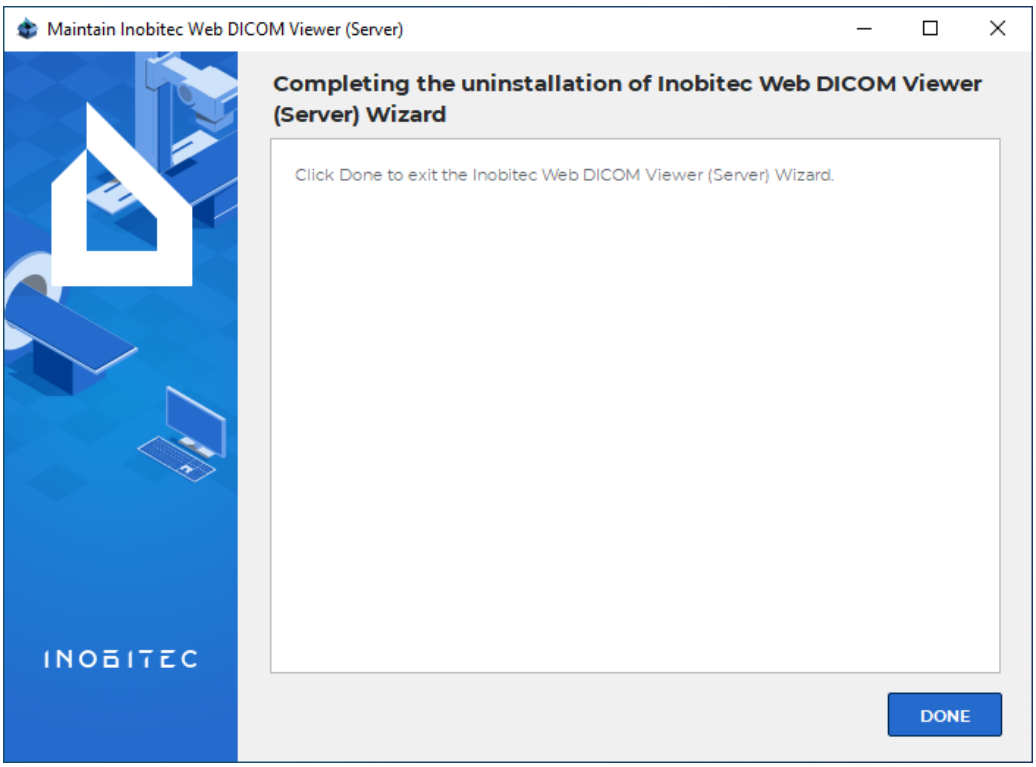


Figure 2.19: Completing the deinstallation procedure

3. In the window for confirmation of the deinstallation procedure (Fig. 2.18) click **UNINSTALL**.

4. In the window for completing the deinstallation procedure (Fig. 2.19) click **DONE**.

2.3.2 Uninstalling the Distributed Program Build

To uninstall the Web DICOM Viewer, delete the directory containing the program on each node. If the server is running as a service, stopping and removing the service is performed using the standard tools of the operating system.

2.4 Installing the Program on Linux

The Web DICOM Viewer supports **SQLite** and **PostgreSQL** DBMS.

- **SQLite** is built-in and doesn't have to be preinstalled;
- **PostgreSQL** (version 9.6 or superior) has to be preinstalled.

Attention! PostgreSQL DBMS must be activated before the Web DICOM Viewer installation procedure begins.

The Web DICOM Viewer for Linux is supplied as a DEB package containing the program and all the libraries required for its work.
The name of the package has the following format: **inobitec-web-dicom-viewer-server-<server version>-<suffix>-<operating system>-<architecture (bitness>-<build type>.deb**.

Below, you will find some names of the Web DICOM Viewer DEB packages for installation on Linux.
inobitec-web-dicom-viewer-server-2.10.2-rW-ubuntu-20.04-x64-unified.deb
inobitec-web-dicom-viewer-server-2.10.2-rW-ubuntu-20.04-x64-distributed.deb

DEB packages with unified and distributed suffixes cannot be installed on the same node.

Installation of a new package will lead to deletion of the previous one. This is due to the fact that both the packages have the same paths for installation and location of the configuration files: */etc/inobitec-web-dicom-viewer-server*.

To install a DEB package:

1. Update the local list of the available software packages from the repositories. To do this, in the terminal, run the command (example for Ubuntu OS):
   sudo apt update

2. Go to the directory with the file and start its installation:
   cd /path/to/DEB package
   sudo apt install ./<name_of_DEB_ package>

   Attention! It is absolutely necessary to use the **./** prefix before the file name. In this case, **apt** will be searching for the package locally, not in some distant repositories.

3. After the DEB package is installed, update the system services. To do this, in the terminal, run the command (example for Ubuntu OS):
   sudo systemctl daemon-reload

When the DEB package is being installed, the **inobitec-web-viewer** user is created, in whose name the server will be launched.

The file structure after installation is as follows:

- configuration files — */etc/inobitec-web-dicom-viewer-server*;
- web client directory — */opt/inobitec-web-dicom-viewer-server/html*;
- directory with log files — */var/log/inobitec-web-dicom-viewer-server* (write access is given to the **inobitec-web-viewer** user).

To view the information on the installed package, execute the following:
apt show inobitec-web-dicom-viewer-server-unified (for the unified build);
or
apt show inobitec-web-dicom-viewer-server-distributed (for the distributed build).

2.4.1 Setup and Launch of the Unified Build of the Program with SQLite DBMS

1. Install the DEB package in compliance with the information provided in Section 2.4. **Attention!** For the unified build installation, use the DEB package with the **unified** suffix.
2. Go to the directory with configuration files by executing:
   cd /etc/inobitec-web-dicom-viewer-server
   Edit the *control.json* and *storage.json* files modifying the node ports and other parameters (optional). The parameter values in the configuration files can be left unchanged.
3. Edit the path to the storage folder and SQLite database in the *renderserverconfig.json* file (optional). The parameter can be applied by default.
4. Launch the server by running the command (example for Ubuntu OS):
   sudo systemctl start inobitec-render-server
5. To enable automatic activation of the server after the Operating System launch, execute the following:
   sudo systemctl enable inobitec-render-server
6. To stop the server, execute the following:
   sudo systemctl stop inobitec-render-server

2.4.2 Setup and Launch of the Unified Build of the Program with PostgreSQL DBMS

Before installing the Web DICOM Viewer, make sure that the PostgreSQL DBMS is installed on the server.

1. Install the DEB package in compliance with the information provided in Section 2.4. **Attention!** For the unified build installation, use the DEB package with the **unified** suffix.

2. Go to the directory with configuration files by executing:
   cd /etc/inobitec-web-dicom-viewer-server
   Edit the *control.json* and *storage.json* files modifying the node ports and other parameters (optional). The parameter values in the configuration files can be left unchanged.

3. Prepare the configuration file for work with PostgreSQL. To do this, copy the PostgreSQL settings from the *renderserverconfig-postgresql.json* file to the *renderserverconfig.json* configuration work file by executing:
   sudo cp renderserverconfig-postgresql.json renderserverconfig.json

4. Edit the new *renderserverconfig.json* file specifying the values for the following database parameters:

   - **driver** — database driver name (default value: *QPSQL*);
   - **name** — database name (default value: *webviewer*);
   - **host** — database server address (default value: *localhost*);
   - **port** — database server port (default value: *5432*);
   - **user** — database user name (default value: *webviewer*);
   - **password** — database user password (default value: *webviewer*).

5. Create a database and a database user by executing the following:
   sudo -u inobitec-web-viewer /opt/inobitec-web-dicom-viewer-server/bin/RenderServer -c /etc/inobitec-web-dicom-viewer-server/renderserverconfig.json --create-db-user --create-database --master-db-user <master_db_name> --master-db-password <master_db_password>

   Attention! You have to provide the username of the database user in the command for the database creation. Otherwise, the log file will be created in the name of the *root* system user and the service will not be able to record events in the log file.

6. Launch the server by running the command (example for Ubuntu OS):
   sudo systemctl start inobitec-render-server

7. To enable automatic activation of the server after the Operating System launch, execute the following:
   sudo systemctl enable inobitec-render-server

8. To stop the server, execute the following:
   sudo systemctl stop inobitec-render-server

2.5 Installation of the Distributed Build of the Program on Linux OS

For the distributed build of the Web DICOM Viewer, it is recommended to use PostgreSQL as the DBMS.

2.5.1 Installation and Launch of the Control Node

1. In the system provided for the control node, install the DEB package in compliance with the information provided in Section 2.4. **Attention!** For the distributed build installation, use the DEB package with the **distributed** suffix.

2. Go to the directory with configuration files by executing:
   cd /etc/inobitec-web-dicom-viewer-server
   Edit the *control.json* file modifying the node port and other parameters (optional). The parameter values in the configuration file can be left unchanged.

3. Prepare the configuration file for work with PostgreSQL. To do this, copy the PostgreSQL settings from the *controlnodeconfig-postgresql.json* file to the *controlnodeconfig.json* configuration work file by executing:
   sudo cp controlnodeconfig-postgresql.json controlnodeconfig.json

4. Edit the new *controlnodeconfig.json* file specifying the values for the following database parameters:

   - **driver** — database driver name (default value: *QPSQL*);
   - **name** — node database name (default value: *control*);
   - **host** — database server address (default value: *localhost*);
   - **port** — database server port (default value: *5432*);
   - **user** — database user name (default value: *control*);
   - **password** — database user password (default value: *control*).

5. Create a database and a database user by executing the following:
   sudo -u inobitec-web-viewer /opt/inobitec-web-dicom-viewer-server/bin/ControlNode -c /etc/inobitec-web-dicom-viewer-server/controlnodeconfig.json --create-db-user --create-database --master-db-user <master_db_name> --master-db-password <master_db_password>

   Attention! You have to provide the username of the database user in the command for the database creation. Otherwise, the log file will be created in the name of the *root* system user and the service will not be able to record events in the log file.

6. Launch the control node by running:
   sudo systemctl start inobitec-control-node

7. To enable automatic activation of the node after the Operating System launch, execute the following:
   sudo systemctl enable inobitec-control-node

8. To stop the node operation, execute the following:
   sudo systemctl stop inobitec-control-node

2.5.2 Installation and Launch of the Storage Node

1. In the system provided for the storage node, install the DEB package in compliance with the information provided in Section 2.4. **Attention!** For the distributed build installation, use the DEB package with the **distributed** suffix.

2. Go to the directory with configuration files by executing:
   cd /etc/inobitec-web-dicom-viewer-server
   Edit the *storage.json* file modifying the node port and other parameters (optional). The parameter values in the configuration file can be left unchanged.

3. Prepare the configuration file for work with PostgreSQL. To do this, copy the PostgreSQL settings from the *storagenodeconfig-postgresql.json* file to the *storagenodeconfig.json* configuration work file by executing:
   sudo cp storagenodeconfig-postgresql.json storagenodeconfig.json

4. Edit the new *storagenodeconfig.json* file specifying the values for the following database parameters:

   - **storage** — path to the storage directory (default value: */var/lib/inobitec-web-viewer/InobitecWebViewerStorage.postgres*);
   - **driver** — database driver name (default value: *QPSQL*);
   - **name** — node database name (default value: *storage*);
   - **host** — database server address (default value: *localhost*);
   - **port** — database server port (default value: *5432*);
   - **user** — database user name (default value: *storage*);
   - **password** — database user password (default value: *storage*).

5. Create a database and a database user by executing the following:
   sudo -u inobitec-web-viewer /opt/inobitec-web-dicom-viewer-server/bin/StorageNode -c /etc/inobitec-web-dicom-viewer-server/storagenodeconfig.json --create-db-user --create-database --master-db-user <master_db_name> --master-db-password <master_db_password>

   Attention! You have to provide the username of the database user in the command for the database creation. Otherwise, the log file will be created in the name of the *root* system user and the service will not be able to record events in the log file.

6. Launch the storage node by running:
   sudo systemctl start inobitec-storage-node

7. To enable automatic activation of the node after the Operating System launch, execute the following:
   sudo systemctl enable inobitec-storage-node

8. To stop the node operation, execute the following:
   sudo systemctl stop inobitec-storage-node

2.5.3 Installation and Launch of the Rendering Node

1. In the system provided for the rendering node, install the DEB package in compliance with the information provided in Section 2.4. **Attention!** For the distributed build installation, use the DEB package with the **distributed** suffix.

2. Launch the rendering node by running:
   sudo systemctl start inobitec-render-node

3. To enable automatic activation of the node after the Operating System launch, execute the following:
   sudo systemctl enable inobitec-render-node

4. To stop the node operation, execute the following:
   sudo systemctl stop inobitec-render-node

2.6 Peculiarities of Running the Program

The peculiarities of running the program on Windows and Linux operating systems when the current working directory is not the installation directory are described below.

2.6.1 Peculiarities of Running the Program on Windows and Linux Operating Systems

If the current working directory is not the installation directory, then for the control node and storage nodes, you need to specify the configuration file for launching. If no such file is specified, then the default value is used as the configuration file name. There is a special value for each executable file. If the specified path to the configuration file is relative (which also applies to the default path), launching the program from an arbitrary directory will, in most cases, cause problems with opening the configuration file. And since the presence of a configuration file is mandatory for the specified executables, this will lead to a launch error. Therefore, in such cases, the path to the configuration file must always be explicitly specified (using the *-c* or *--config* flag) and must be either absolute or correct for the current working directory.

The unified build and all types of nodes write a log file. For each executable file, the default log file name is specific, but it is always assigned with consideration to the current directory. The current directory may be inaccessible or not intended for writing. Therefore, in most cases, it is better to specify the path to the log file each time (using the *-l* or *--log* flag).

2.7 Uninstalling the Program on Linux OS

To remove the Web DICOM Viewer while preserving the configuration files, run:
sudo apt remove inobitec-web-dicom-viewer-server-unified (for the unified build)
or
sudo apt remove inobitec-web-dicom-viewer-server-distributed (for the distributed build).

To remove the Web DICOM Viewer along with all configuration files, run:
sudo apt purge inobitec-web-dicom-viewer-server-unified (for the unified build)

or
sudo apt purge inobitec-web-dicom-viewer-server-distributed (for the distributed build).

2.8 Command Line Parameters

General parameters for all executable files:

-?, **-h**, **--help**: display the list of command-line parameters and terminate execution. The **-?** parameter is only applicable for Windows OS;

-v, **--version**: display the version of the executable file and terminate execution;

-l, **--log <file name>**: specifies the path for the log file;

--log-level <level>: sets the logging level. <level> can take the following values: **error**, **warning**, **info**, **debug**;

--log-to-console: enables duplication of log messages to the console;

--no-log-to-console: disables duplication of log messages to the console;

--log-flush: perform the flush of messages to the magazine file after each event;

--no-log-flush: do not flush the log file after each message;

--log-high-precision-time: ensures microsecond precision for the timestamps of messages in the log file;

--no-log-high-precision-time: disables microsecond precision for the timestamps of messages in the log file.

Parameters shared by the unified build, control node, and storage nodes:

-c, **--config <file name>**: specifies the path to the configuration file.

Parameters for database maintenance operations for the unified build, control node, and storage nodes:

--check-database: when this key is used, the server determines the current state of the database and one of the following exit codes is provided:

0 — the database is updated and ready to use;

1 — Internal error. The error may be due to invalid database configuration parameters or a problem with the DBMS driver initialization;

2 — database connection error. The database does not exist or it is impossible to establish a connection with this database with the parameters provided. It is recommended to create a new database;

3 — the database is empty. The database exists, but does not contain any tables. New tables must be created for this database;

4 — invalid database scheme. The database contains tables, but the relevant tables are missing, or the tables do not contain the relevant fields. You need to choose a different database. To do this, change one of the following parameters: the name of the database, the host, or the port;

5 — outdated database contents. The contents of some tables may be outdated. You need to choose a different database. To do this, change one of the following parameters: the name of the database, the host, or the port;

6 — unsupported database version. The database corresponds to the new program version which is incompatible with the current server version. You need to choose a different database. To do this, change one of the following parameters: the name of the database, the host, or the port;

7 — outdated database version. You need to upgrade the database to the new version.

Attention! The **--check-database** key is incompatible with the following database maintenance keys: **--create-database**, **--create-db-user**, **--drop-database**, **--update-database**, **--create-db-scheme**. When incompatible keys are provided, database maintenance commands are not executed.

--drop-database: delete the database of the node or unified build, the name of which is specified in the configuration file, or delete the database file if SQLite is used;

--create-db-user: create a database user for the node or unified build, whose name and password are specified in the configuration file (not applicable to SQLite). The name and the password of a user with administrative privileges must be specified using the **--master-db-user** and **--master-db-password** keys;

--create-database: create a database for the node or unified build. The name of the database (for SQLite the path to the database file) is specified in the configuration file. Grant all the privileges for the created database to the database user whose name is specified in the configuration file. The name and the password of a user with administrative privileges must be specified using the **--master-db-user** and **--master-db-password** keys (except for SQLite). It may also be necessary to specify the database for administrative connection using the **--master-db-name** key (e.g., DSN in the case of ODBC);

--master-db-user <name>: the name of the user with administrative privileges for performing operations on behalf of such a user;

--master-db-password <password>: the password of the user with administrative privileges;

--master-db-name <name>: the name of the database for operations with administrative privileges, if required;

--update-database: update the database for the node or the unified build (the name is specified in the configuration file), or the database file, if SQLite is used;

--create-db-scheme: create a scheme in an empty database. When the key is provided, the database is checked for the presence of tables. If there are any tables in the database, the scheme is not created.

Parameters for the rendering node and unified build:

--device <index>: the number of the rendering device for 3D/MPR reconstructions from the list of devices provided to the log file or to the console when the rendering node or unified build is launched. By default, the rendering device with the highest performance is selected;

--port <port>: the tcp port for http listener;

--render-bandwidth <cps>: the limit on the maximum bandwidth for rendering requests per one render context in bytes per second. The value is unlimited by default;

--max-thread-pool-threads <count>: the number of threads used for parallelizing rendering operations. The default value is 4;

--no-image-cache: do not cache images in case of reopening in other contexts;

--no-volume-cache: do not cache 3D/MPR reconstructions in case of reopening in other contexts;

--no-cache: do not cache images or 3D/MPR reconstructions in case of reopening in other contexts.

Chapter 3

Administrator Web Console

3.1 Access to the Administrator Web Console

Administrator Web Console is available by the
http://<ip address>:<port>/admin,
where **port** is the port with the Control Node (for the distributed build) or the RenderServer port (for the unified build). By default, it is port 8090.

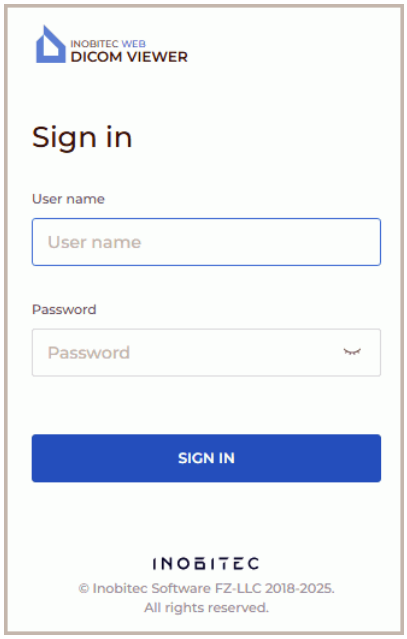


Figure 3.1: Logging into the administrator web console

In the pop-up box (Fig. 3.1) print the user name (login) and password, and then click the **SIGN IN** button. The entered password is hidden and displayed as dots. To make the password visible in the input field, click on the icon representing a closed eye.
The following credentials are used by default:

User name: admin
Password: admin

Main menu of the administrator web console of the Web DICOM Viewer is swown in Fig. 3.2.

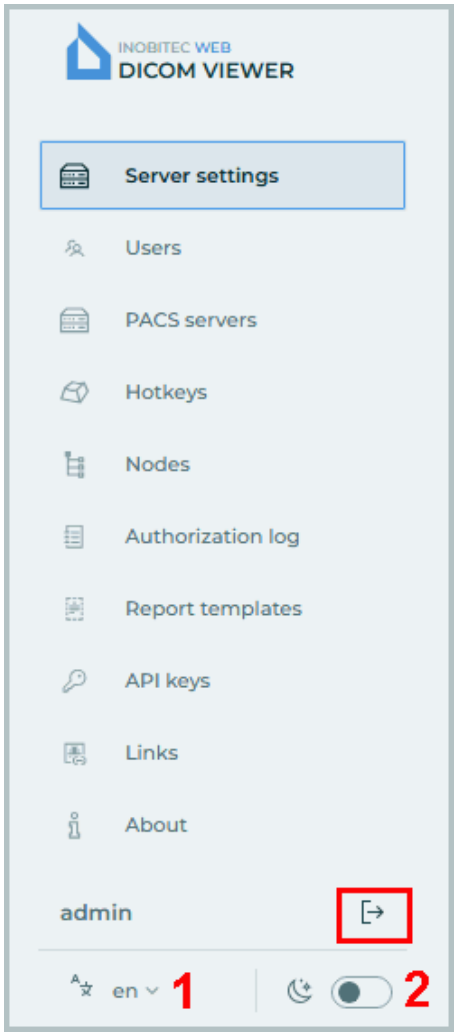


Figure 3.2: Main menu of the administrator web console

To change the interface language, click the language selection button in the lower left corner (marked with number «1» in Fig. 3.2). To change the page background click the switch marked with number «2» in Fig. 3.2.

To log out from the administrator web console, click the **Sign out** button on the main menu (see the red box in Fig. 3.2).

The administrator can set the default interface language for all the users. To do this, follow these steps:

1. Navigate to the directory where the Web DICOM Viewersoftware package is installed. Go to the **html** directory.

2. In the **.config.json** configuration file, change the value of the *language* parameter. If the parameter is missing in the configuration file, it needs to be added. The admissible

values are:

- **en** — sets English as the interface language for all users;
- **pt** — sets Portuguese (Brazilian) as the interface language for all users.

3. Save the edited configuration file.

The user can change the interface language set by the administrator by default.

3.2 Server settings

On the **Server settings** page (Fig. 3.3) activates the license and changes the administrator password.

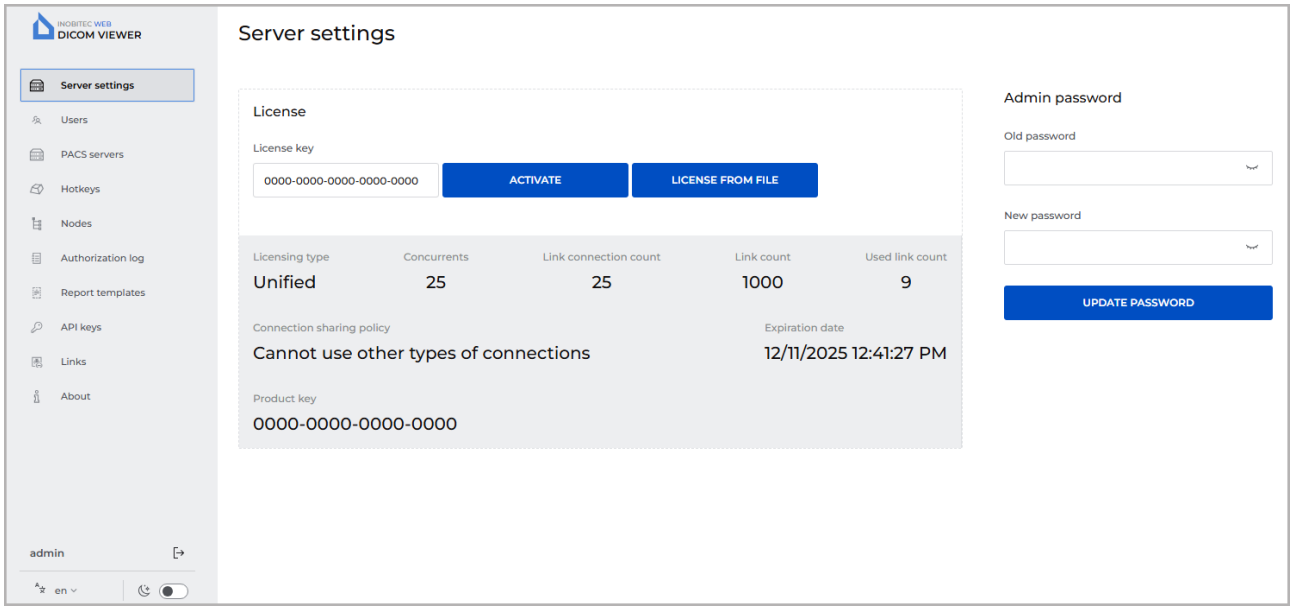


Figure 3.3: **Server settings** *page*

3.3 Licensing. License activation

On the **Server settings** page (Fig. 3.3) activates the license.

3.3.1 Trial Period

The trial period allows the user to explore the capabilities of the Web DICOM Viewer by using it without any functional limitations for 14 days. A trial license can only be activated by the user if it has not been previously issued for the Web DICOM Viewer with the current product code. The product code is displayed on the **Server settings** page (see Section 3.3.2).

Ttrial period limitations:

- only the unified server edition using the **SQLite** DBMS is available;

- the duration of a working session is limited to 1 hour.

To activate a trial license, click the **ACTIVATE TRIAL LICENSE** button on the **Server settings** page. After successful activation of the trial license, the **Server settings** page will display the license type and the trial period expiration date.

Attention! Self-activation of the trial period through the program interface is not available if the Web DICOM Viewer is:

- **configured to use PostgreSQL DBMS;**
- **installed in a virtual environment.**

To obtain a trial period for the program that uses PostgreSQL DBMS or is installed in a virtual environment, you must contact the support service at **support@inobitec.com**.

After 14 days, you will no longer be able to use the Web DICOM Viewer, and to continue working, you will have to purchase a license.

To purchase a license, contact the sales department at the email address **market@inobitec.com**.

3.3.2 License activation

On the **Server settings** page, the product key is displayed in the **Product key** field and the license key is to be provided.

You can activate the license in two ways:

1. **With a license key**. Enter the license key in the **License key** field and click the **ACTIVATE** button.
2. **With a license from file**. Click the **License from file** button. In the dialog box that opens, select the license file. **Attention! With this activation method, not all server modes may be available.** For more information, please contact technical support at **support@inobitec.com**.

In the **Licensing type** field, you see the information about the current server mode. It depends on the license key type:

- if the user has activated a trial license, the **Licensing type** field displays **Trial** (see Section 3.3.1);
- if the server license key is for a personal user license, in the **Licensing type** field you will see the words **Personal** (see Section 3.3.3);
- if the server license key is for a concurrent license, in the **Licensing type** field you will see the words **Concurrent**. In the **Concurrents** field, you see the maximum number of users who can work with the system simultaneously. The total number of the system users may be greater than the number of concurrent connections (see Section 3.3.4);
- if the server license key combines the capabilities of a personal and a concurrent licens with additional features, the **Licensing type** field displays **Unified** (see Section 3.3.5).

3.3.3 Personal User Licenses

A personal user license is registered in the user's account on the **Users** page (see Section 3.5.2).

The license key activated for a certain user creates one personalized connection providing for continuous access to the system. The user with a personal license has constant access to the system.

3.3.4 Concurrent licenses

A concurrent license is a license limiting the number of users who can work in the system simultaneously.

The Web DICOM Viewer server automatically provides an unoccupied connection to the user when he/she logs in. The connection provided belongs to the shared unoccupied connection pool. The shared connection pool is a list of unoccupied concurrent connections. The connection stays active during the work session, which ends after the user logs out.

When a user opens a study from a link it uses a competitive user connection.

Attention! If there are not enough concurrent connections, the user will be able to work in the system only after another user log out and an unoccupied connection appears.

3.3.5 Unified User Licenses

A unified license combines the capabilities of a personal and a concurrent license with a range of additional features.

After activating a unified license key, the **Server settings** page (Fig. 3.3) displays information on the current server mode.

In the **Concurrents** field, you see the maximum number of users who can work with the system simultaneously. The total number of the system users may exceed the number of concurrent connections.

In the **Link connection count** field you see the maximum number of connections for opening a study simultaneously via a link.

In the **Link count** field, the number of links that can be signed by the server is shown. The links signed by the server can be opened by an unlimited number of users simultaneously (regardless of the number of concurrent link connections specified in the **Link connection count** field).

The number of server-signed links is displayed in the **Used link count** field. The number of links that can still be signed by the server is determined as the difference between the total number of links and the number of used links.

In the **Connection sharing policy** field, the policy for sharing concurrent user connections and opening studies via links is described:

- **Cannot use other types of connections**: no sharing of concurrent user connections and opening studies via links;
- **Connections by link can use users connections**: connections for opening studies via links can use concurrent user connections;
- **Users connections can use connections by link**: concurrent user connections can use connections for opening studies via links.

In the **Expiration date** field, the date and time of the license expiration are displayed.

3.4 Changing the administrator password

On the **Server settings** page (Fig. 3.3) you can change the administrator password.

To change the Web DICOM Viewer administrator password, enter the current password in the **Old password** field and the new password in the **New Password** field. A null value is inadmissible! Passwords are hidden when entered. To display the password, press the icon in the shape of an eye. To save the new administrator password click the **UPDATE PASSWORD** button.

3.5 User settings

On the **Users** page (Fig. 3.4) you can manage users.

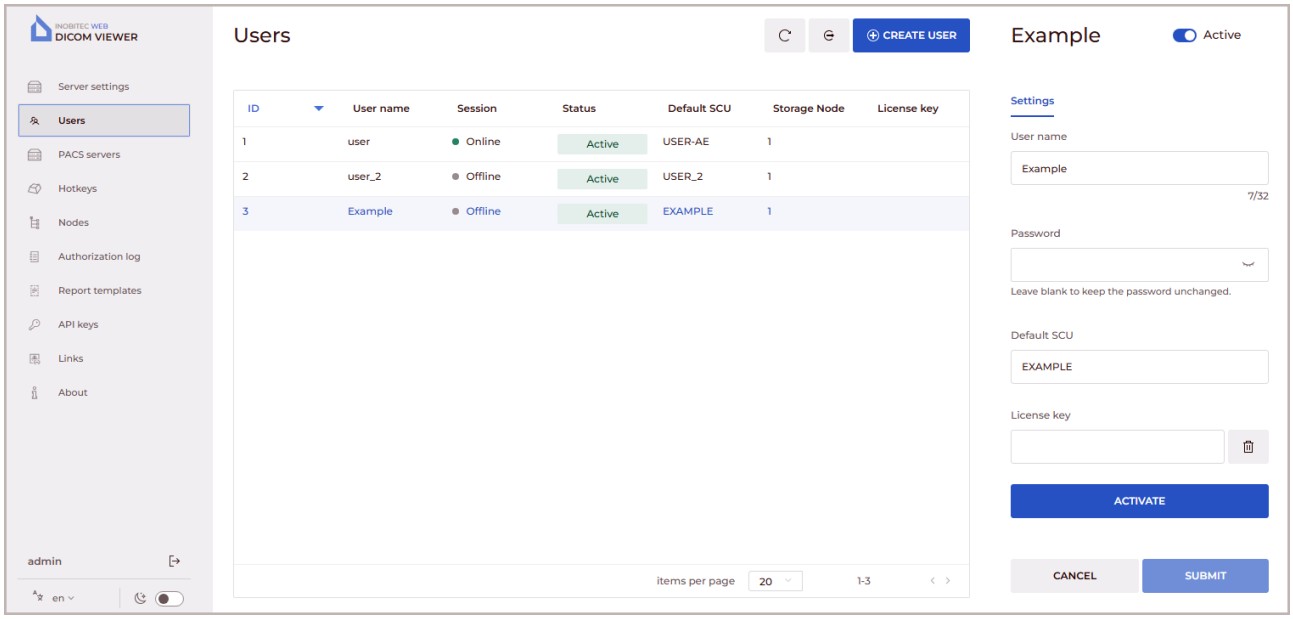


Figure 3.4: **Users** *page*

The table shows the users parameters.

For details on the parameters display options, see Section 3.14.1. The users may be sorted by the values provided in one of the columns (see Section 3.14.2).

At the bottom of the page, you see the range of users for the current page and the total number of users. To go to the next page of the list, click the ❯ button; to go to the previous page — click the ❮ button. To go to the last page of the list, click the ❯| button, to return to the first page — click the |❮ button.

On the **items per page** drop-down list, select the number of users to be displayed on a page.

3.5.1 Creating New Users

To add a new user, click the **CREATE USER** button on the **Users** page.

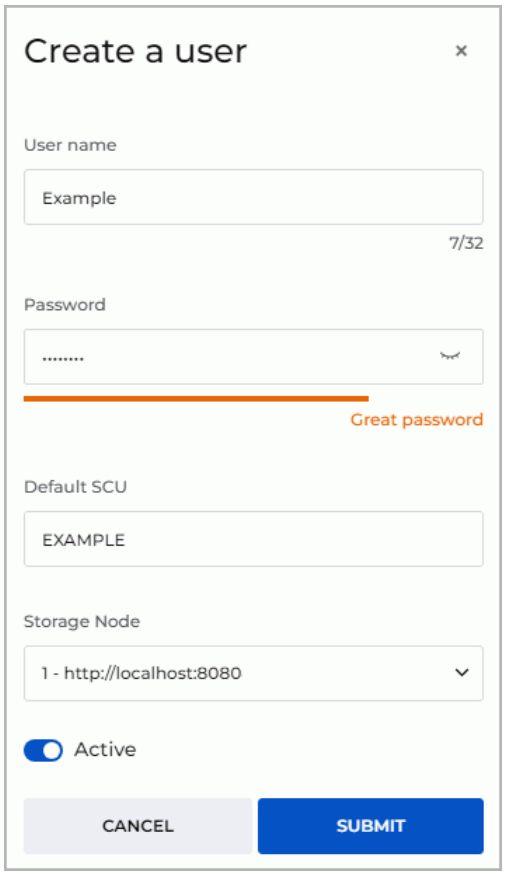


Figure 3.5: Create a user dialog box

Type the user's parameters in the **Create a user** dialog box (Fig. 3.5) paying attention to the following requirements:

- The first character in the **User name** must be a letter. The name may include digits, letters of the Latin alphabet, as well as underscores. The **User name** must be up to 32 characters. No punctuation marks, spaces, quotation marks or special characters are to be used.
- The user's **Password**. The length of the password is not limited. When the password is entered, its strength is indicated. To display the password, press the icon in the shape of an eye;
- The **Default SCU** can contain digits, letters of the Latin alphabet hyphens, and underscores. The SCU length must not exceed 16 characters. Punctuation marks, quotation marks, special characters, and spaces are not allowed;
- from the **Storage Node** dropdown list, select the address of the active Storage Node where the user's DICOM studies will be stored (see Section 3.8). This option is only available for the distributed build of the program.

.

To enable/disable access, move the **Active** switch. The user is active by default.

Click **SUBMIT** to create a new user or **CANCEL** to cancel.

3.5.2 Managing Users in the Personal User Licensing Mode

Select a user in the list on the **Users** page. On the right-hand side of the window, the user's credentials are displayed in the **Settings** area. To edit the user's parameters, type the new data in the respective fields (Fig. 3.6).

The requirements for creating the **User name**, the **Password** and **Default SCU** are provided in Section 3.5.1.

Enter the user's personal license key in the **License key** field and click the **ACTIVATE** button.

To enable/disable access, move the **Active** switch.

Attention! After changing the username, studies stored in the storage and on the user's personal PACS servers become inaccessible for opening via previously created links.

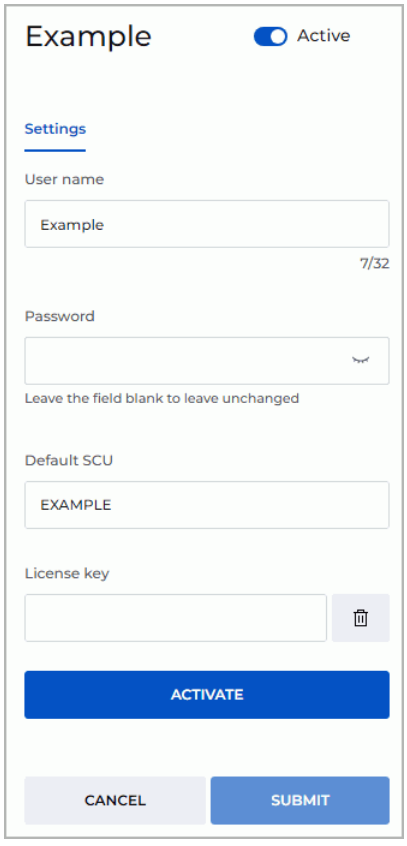


Figure 3.6: Editing the user's settings

When a personal license key is activated for a certain user, one personalized connection is created providing for continuous access to the system.

To delete the user's license key, click the **Remove** button.

Attention! If a license key has been deleted, it cannot be used in future.

Click **SUBMIT** to apply the settings or **CANCEL** to cancel.

3.5.3 Managing Users in the Concurrent Licensing Mode

Select a user in the list on the **Users** page. On the right-hand side of the window, the user's credentials are displayed in the **Settings** area. To edit the user's parameters, type the new data in the respective fields (Fig. 3.7).

The requirements for creating the **User name**, the **Password** and **Default SCU** are provided in Section 3.5.1.

Click **SUBMIT** to apply the settings or **CANCEL** to cancel.

To enable/disable access, click the **Active** switch.

Attention! After changing the username, studies stored in the storage and on the user's personal PACS servers become inaccessible for opening via previously created links.

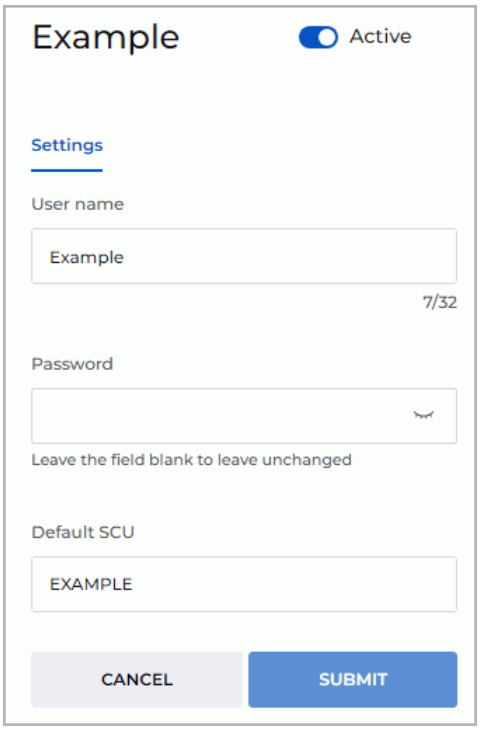


Figure 3.7: Editing the user's settings

The total number of the system users may be greater than the number of concurrent connections. If all the concurrent connections are busy, a message will pop up when the user tries to enter the system.

3.5.4 Managing Users in the Unified Licensing Mode

User management in the **Unified** licensing mode is performed similarly to user management in the personal licensing mode and is described in Section 3.5.2.

Attention! After the username is changed, the studies stored in the user's storage and on the user's personal PACS servers cannot be opened via previously created links.

3.5.5 User sessions

On the **Users** page you can manage user sessions (Fig. 3.4).

To stop the session, click the **Close the user session** button.

To update the session list, click the **Update user list** button.

3.6 Configuring the Connection to PACS Servers

On the **PACS servers** page, you can configure common connections to PACS servers and applications with which data exchange via the DICOM protocol becomes possible. Common connections to PACS servers are available to all the Web DICOM Viewer users.

The table displays the parameters of PACS servers connected to the Web DICOM Viewer. For more details on setting up the display of the table parameters, see Section 3.14.1. The parameters can be sorted in the table by the values in one of the displayed columns (see Section 3.14.2).

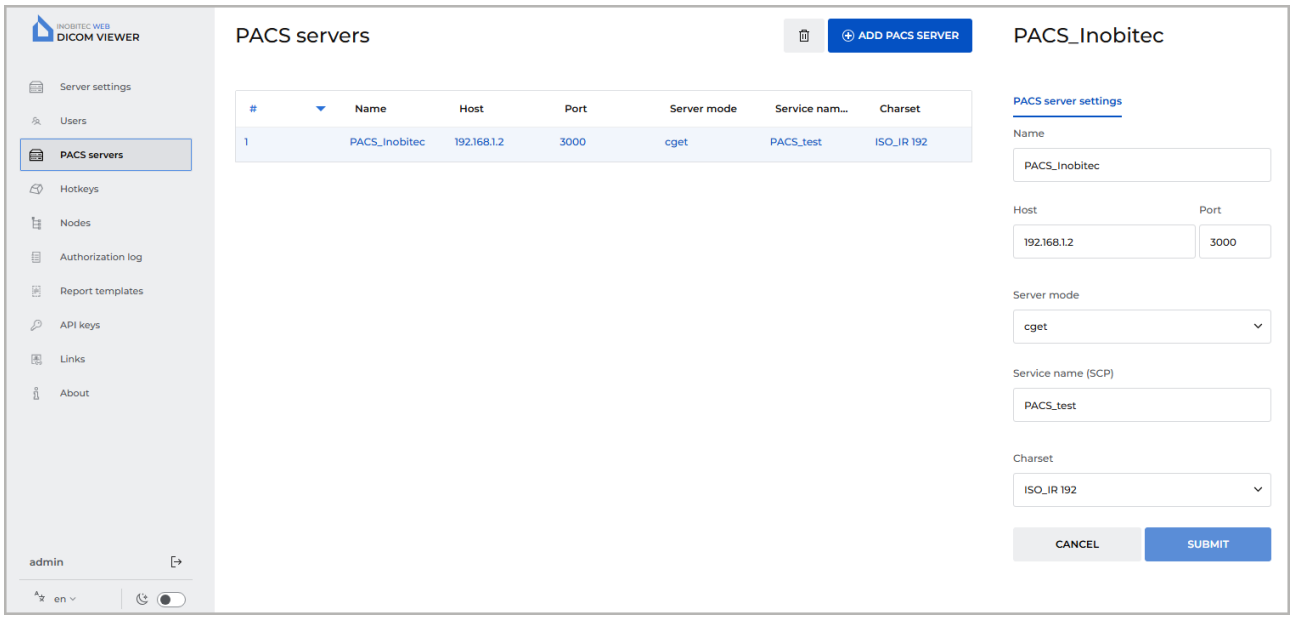


Figure 3.8: **PACS servers** *page*

To add a new connection to a PACS server, follow these steps:

1. Click the **ADD PACS SERVER** button.
2. In the **Create a PACS server** dialog box that pops up (Fig. 3.9), fill in the following fields:
   - in the **Name** field, type the name of the PACS server;
   - in the **Host** field, type the IP address of the PACS server without spaces;
   - in the **Port** field, enter the port value in the range from 0 to 65535;

- on the **Server mode** dropdown list, select the desired mode;
- in the **Service name (SCP)** field, type the name of the server where the PACS server is installed;
- on the **Charset** dropdown list, select the encoding supported by the PACS server.

3. Click the **SUBMIT** button to save the information and close the window, or **CANCEL** to cancel.

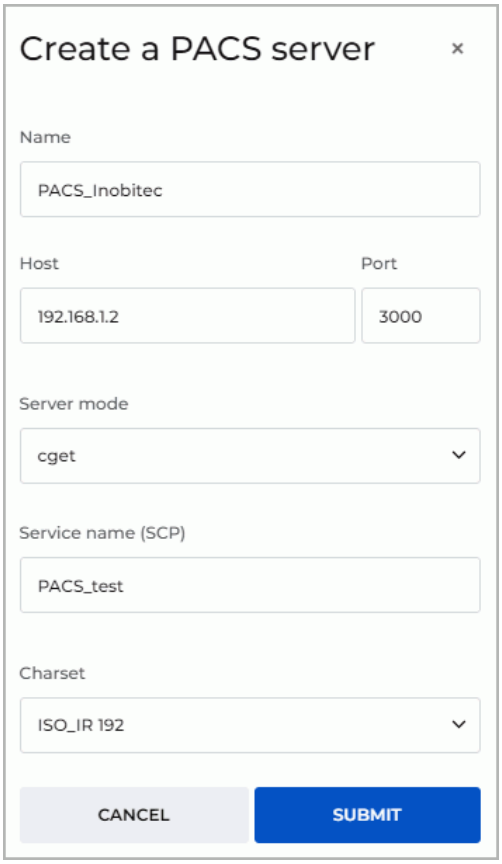


Figure 3.9: Creating a new PACS server

To modify the connection parameters of any PACS server, perform the following:

1. Select the PACS server from the list and edit its parameters in the **PACS server settings** area located on the right-hand side of the page (Fig. 3.8).

2. Click the **SUBMIT** button to apply the changes, or **CANCEL** to cancel.

Attention! After the PACS server name is changed, the studies stored on that PACS server annot be opened via previously created links.

To delete a connection to a selected PACS server, click the **Delete the PACS server** button  . In the confirmation dialog box click **YES** to delete or **CANCEL** to cancel.

The administrator can edit or delete only common connections available to all users. The functions of editing and deleting personal connections to PACS servers are only available for the users who created those connections.

3.7 Hotkeys

On the **Hotkeys** page you can set the default hotkeys (Fig. 3.10).

The table shows the settings for displaying tools in the program interface and the respective hot keys. For details on customizing parameters display options, see Section 3.14.1. The tools may be sorted in the table by the values shown in one of the columns (see Section 3.14.2).

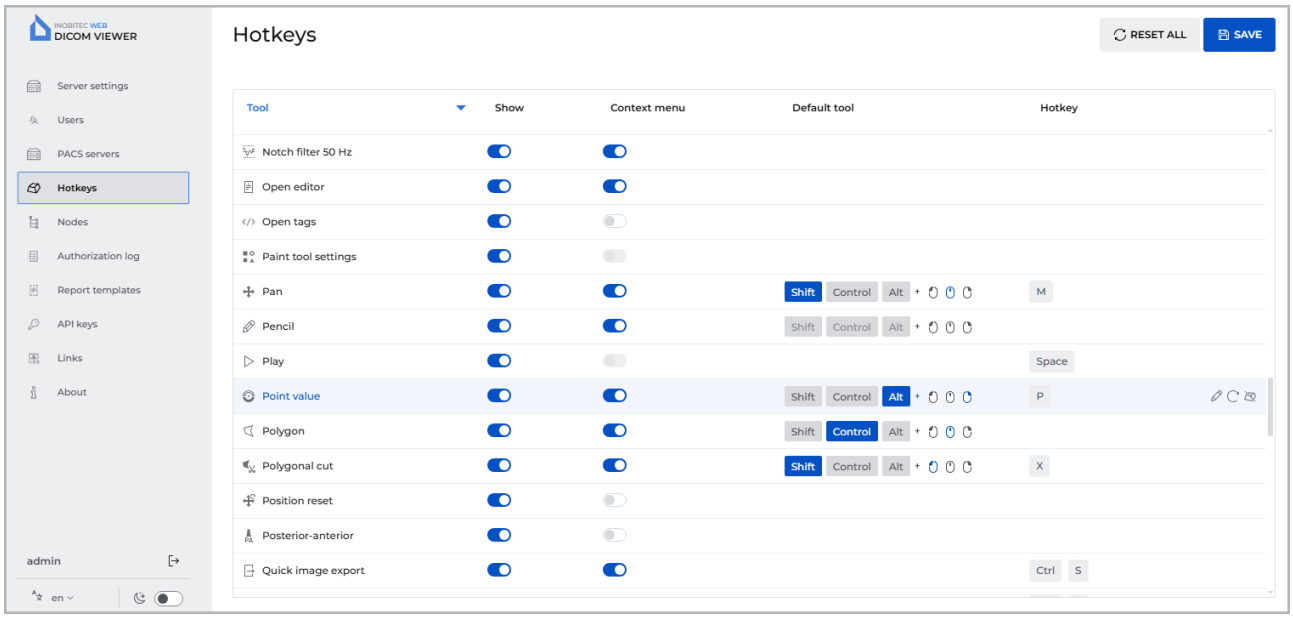


Figure 3.10: **Hotkeys** *page*

If you want a tool or action to be displayed on the toolbar, put the **Show** switch in the right position. If you want to hide the tool, choose the left position of the switch. If you want a tool to be shown on the context menu, put the switch in the **Context menu** column in the right position. If you don't want the tool to be shown, choose the left position of the switch.

In the **Default tool** column, a combination of modifier keys (**Shift**, **Ctrl**, **Alt**) and mouse buttons is assigned to activate the tool when the is combination is pressed. To change or assign a default key combination to a selected tool, follow these steps:

- click on one of the mouse button thumbnails to activate the selected tool with the left, middle, or right mouse button;
- select a modifier key.

The combination of a mouse button and a modifier key must be unique for each tool. If the selected mouse button and modifier key are already used for another tool, they are highlighted in red and a duplicate warning up. The **SAVE** button is deactivated. Assigning a modifier key is not mandatory.

In the **Hotkey** column, you can see the hot keys assigned to the respective tools or actions. To change or assign a hot key for a selected tool or action, put the mouse cursor on the desired line, click the **Update hotkey** button and enter the name of the key in the respective field. If the key is already being used for another action, a warning will pop up.

To restore the default key combination assigned to a tool or action, click the **Reset hotkey** button.

To delete a hot key assigned to a tool, click the **Delete hotkey** button.

Click **SAVE** to save the settings. To restore the default tool display settings and hot keys, click the **RESET ALL** button. In the dialog box that appears, click **YES** to delete or **CANCEL** to cancel.

3.8 Nodes

Attention! The Nodes page is only available for the distributed build of the program.
On the **Nodes** page information on the Web DICOM Viewer nodes which are stored in the Control Node database is provided (Fig. 3.11).

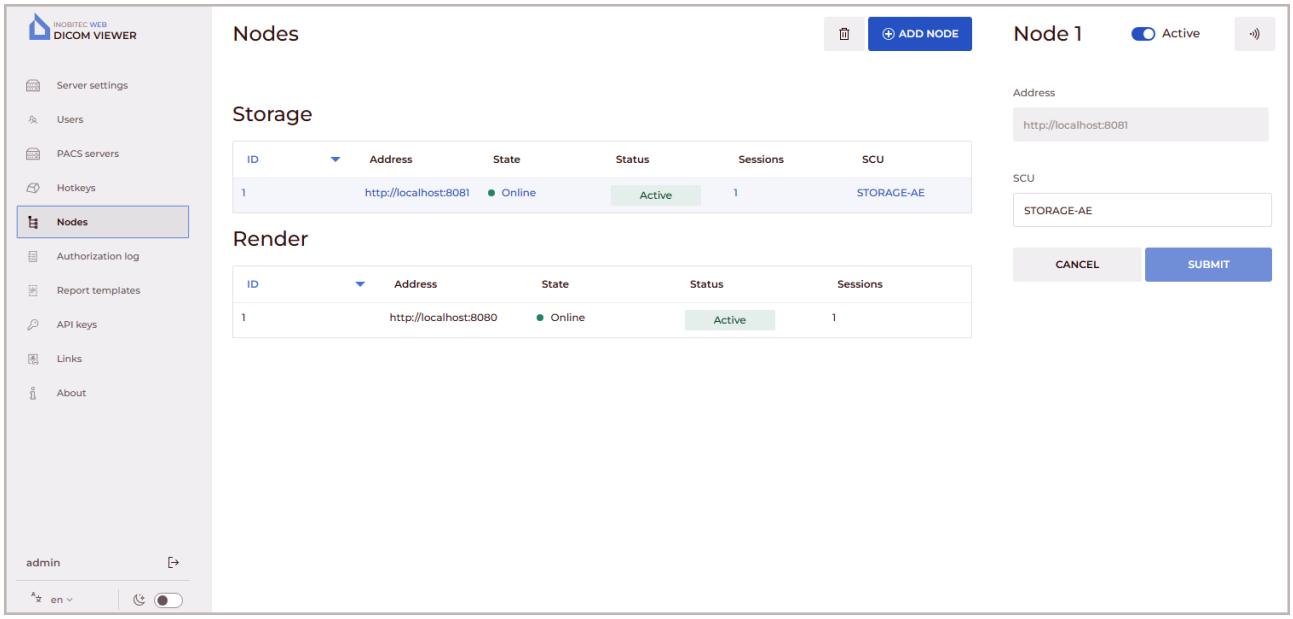


Figure 3.11: **Nodes** *page*

The table for connected nodes shows the following connection parameters:

- the node number in the **ID** column;
- the node address in URL format in the **Address** column;
- the node availability in the **State** column;
- the node activity in the **Status** column;
- the number of active sessions with the node in the **Sessions** column;

- the SCU value for the Storage Node, which is required for opening via a link from a PACS server, in the **SCU** column.

For details on customizing parameters display options, see Section 3.14.1. The nodes may be sorted in the table by the values shown in one of the columns (see Section 3.14.2).

To change the port, IP address or SCU for some node, proceed as follows:

1. If required, edit the configuration file and/or the node service startup parameters.
2. Restart the service of the edited node.
3. In the administrator's web console, go to the **Nodes** page. Select the node whose parameters have been modified. On the right-hand side of the **Nodes** page, edit the parameter values, specifying the current ones. The status of the node may be changed if the selected node is available.
4. Click the **SUBMIT** button, to apply the changes or **CANCEL** to cancel.

To change the status of the selected node (active/inactive), adjust the **Active/Inactive** toggle switch. To check the availability of the selected node, click the **Ping the node** button .

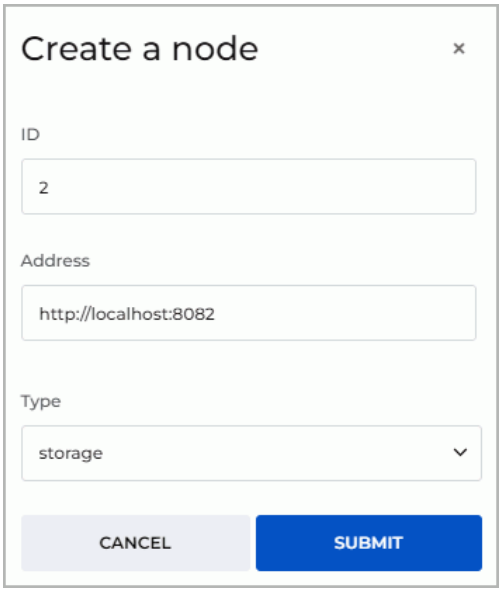


Figure 3.12: Creating a new node

To add a connection:

1. Click the **ADD NODE** button.
2. In the **Create a node** dialog box (Fig. 3.12) set:
   - the node number in the **ID** field. By default, the next consecutive number is entered in the field;

- in the **Address** field the IP address and the port in the following format: *http:///<IP-address>:<port>*;
- the node type by selecting from the **Type** dropdown list.

3. Click **SUBMIT**, to apply changes or **CANCEL** to cancel.

To remove the connection, click the **Delete the node** button.

3.9 Authorization log

On the **Authorization log** page, the data concerning the frequency and duration of the application use are collected. In the table provided, you can see the list of events in the system (authentication, password change etc.).

For details on customizing the list display options, see Section 3.14.1. The events may be sorted in the table by the values shown in one of the columns (see Section 3.14.2).

Besides, the authorization log helps to prevent unauthorized access, as the attempts to enter the system with an invalid user name or password are recorded.

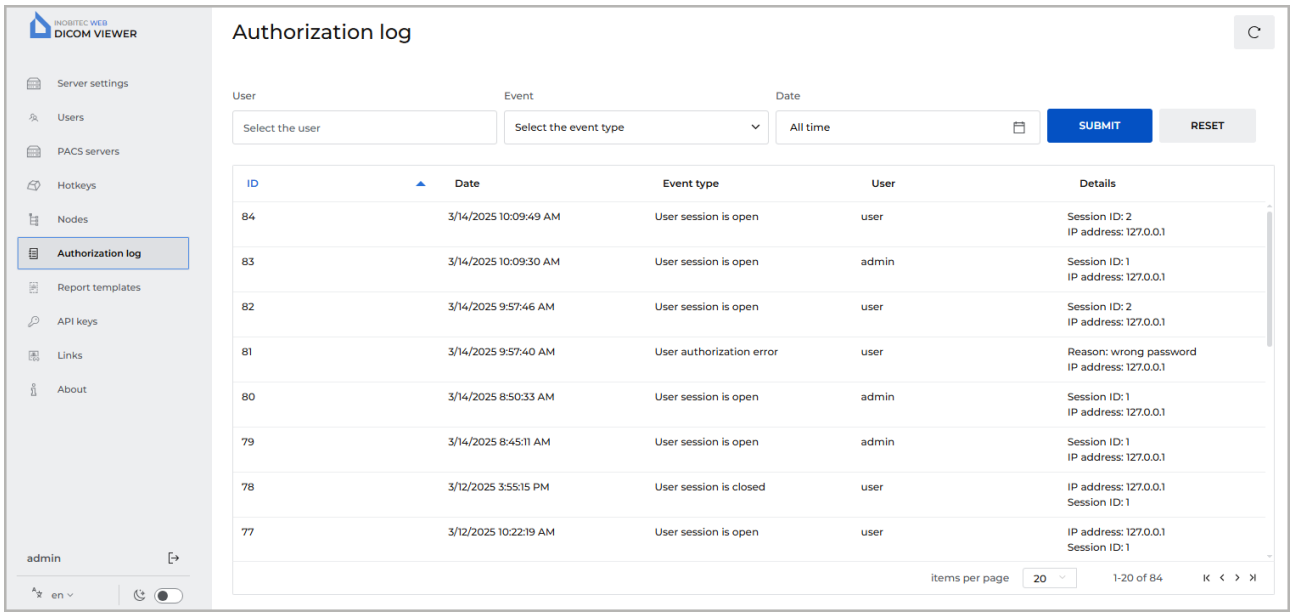


Figure 3.13: **Authorization log** *page*

The authorization log provides the following information:

- **ID** — the unique identifier of the event;
- **Date** — the date and time of the event;
- **Event type** — events that are displayed in the authorization log:
  - **Admin authorization error**;
  - **User authorization error**;

  - **User session is open**;
  - **User session closed**;
  - **Session expired**;
  - **User data has been changed**;

- **User** — the name of the user who performed the action;
- **Details** — additional information about the event:
  - **IP address**;
  - **Old login** (old user login if user settings have been changed);
  - **New login** (new user login if user settings have been changed);
  - **New status** (new user's status);
  - **Old storage node ID**;
  - **New storage node ID**;
  - **Session ID** (the ID of the session during which the event occurred);
  - **Login** (the login used by the user in a failed login attempt);
  - **Password** (the user's password has been changed);
  - **Reason** (event description).

At the top of the authorization log, you see the event filtering panel. The search can be performed by the following parameters:

- **User**;
- **Event**;
- **Date**.

Click **SUBMIT** to search, or **RESET** to reset the specified event filtering parameters. To update the events list, click the **Update events list** button.

At the bottom of the log, you see the range of events for the current page and the total number of events. To go to the next page of the log, click the button; to go to the previous page — click the button. To go to the last page of the log, click the button, to return to the first page — click the button.

On the **items per page** drop-down list, select the number of events to be displayed on the authorization log page. By default, 20 events per page are displayed.

3.10 Report Templates

On the **Report templates** page (Fig. 3.14), the administrator creates, edits, and deletes general report templates. General report templates are displayed on the template list for all program users. The functions of creation, editing, and deletion of general report templates are only available to the administrator.

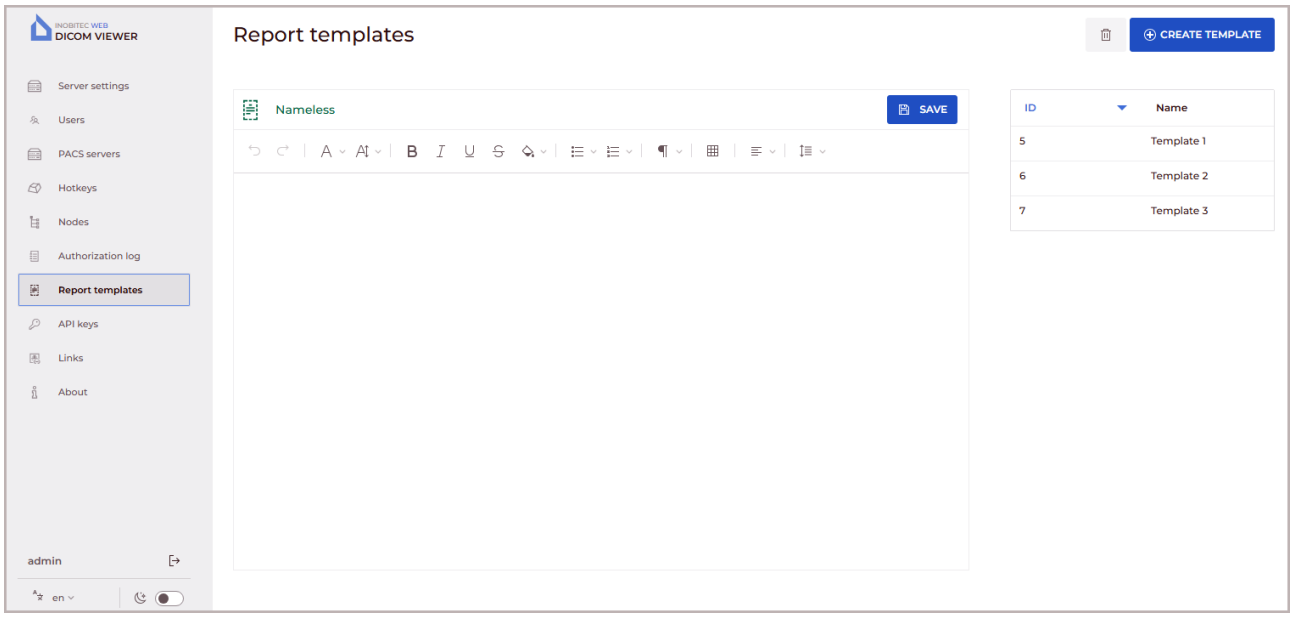


Figure 3.14: **Report templates** *page*

The page features the editor window and the list of general templates. The editor window and text editing tools are described in detail in Section **Editor Window** of the User Manual.

To create a new general report template, click the **CREATE TEMPLATE** button. By default, an empty template named **Nameless** is created. Change the template name and enter the information required. To save the report template, click the **SAVE** button. The saved template is added to the template list.

To open a previously created template for viewing and editing, select the desired template from the list. Edit the template's name and content. Save the edited report template by clicking the **SAVE** button.

To switch to another template, select it from the template list. If there are any unsaved changes in the current template, a dialog box with a warning will pop up when you attempt to switch. In the confirmation dialog, click **SWITCH** to switch to the other template, after which the unsaved data will be lost, or **CANCEL** to cancel the switch.

To delete a selected template from the list, click the **Delete template** button  . In the confirmation dialog that pops up, click **YES** to delete the report template or **CANCEL** to cancel.

3.11 API keys

A special authorization key is required for execution of API requests. On the **API keys** page (Fig. 3.15), the administrator manages access keys.

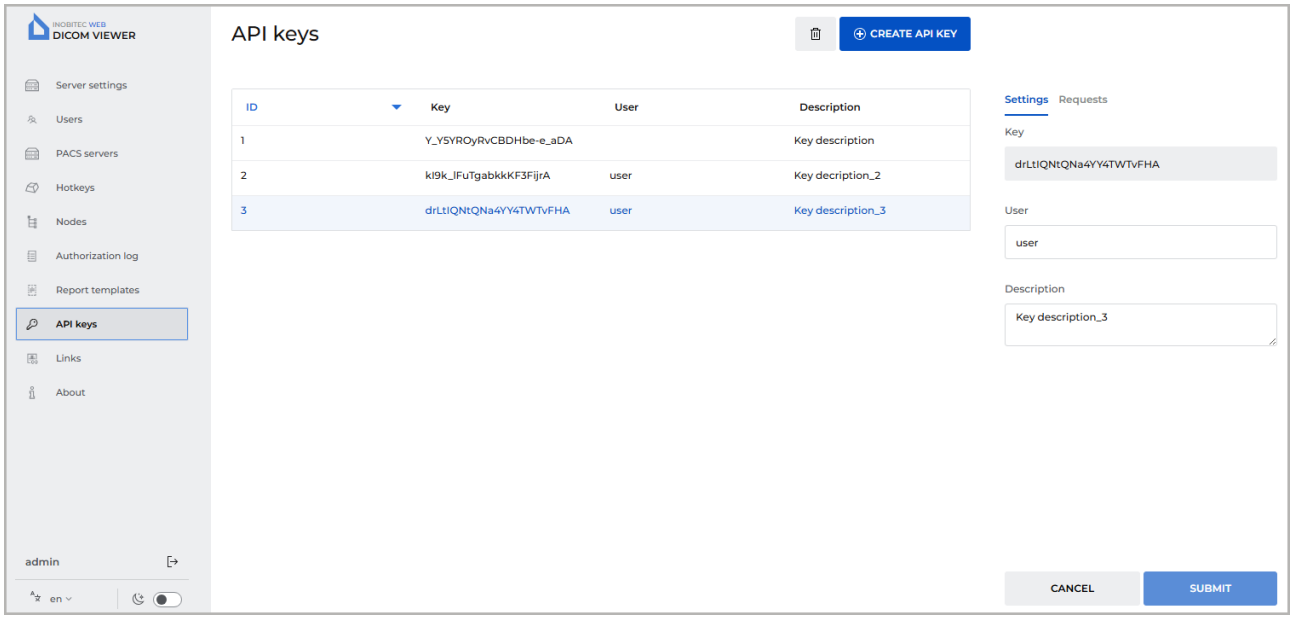


Figure 3.15: API keys page

The table displays the following access key parameters:

- key number in the **ID** column;
- generated access key in the **Key** column;
- the user on whose behalf the requests are executed in the **User** column. If there is no username in the column, the request is executed on behalf of the administrator;
- key description in the **Description** column.

For more details on configuring the display of table parameters, see Section 3.14.1. The parameters can be sorted in the table by the values of one of the displayed columns (see Section 3.14.2).

To create a new API key, follow these steps:

1. Click the **CREATE API KEY** button.
2. In the **Creating an API key** dialog box (Fig. 3.16) that pops up, on the **Settings** tab, fill in the following fields:
    - in the **User** field, enter the username on whose behalf the requests will be executed. Entering a non-existent username is not allowed. To execute requests on behalf of the administrator, leave the **User** field empty;
    - in the **Description** field, add a description of the key if necessary.

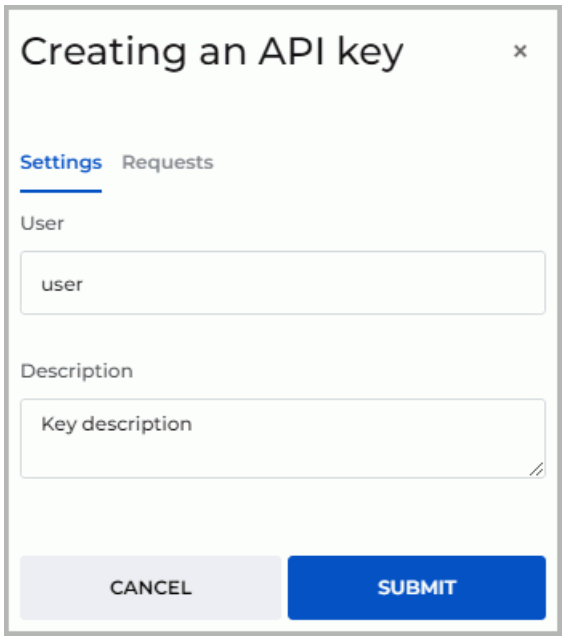


Figure 3.16: The **Settings** *tab of the* **Creating an API key** *dialog box*

3. In the **Creating an API key** dialog box (Fig. 3.17), on the **Requests** tab, do the following:
   - From the **Request** dropdown list, choose an available request. Currently, only the *POST/api/link* request is available;
   - Activate the request by toggling the **Enabled** switch. By default, the request is deactivated;
   - Set key restrictions for this request when creating links. To permit an action, check the respective box, to prohibit an action, uncheck it. By default, all the boxes are checked.
     - The **Creating links to studies in the user storage** option controls the restriction on creating links to studies located in the user's storage;
     - The **Creating links using the user's PACS server** option controls the restriction on creating links to studies located on the user's personal PACS server. From the dropdown list, select the required number of the user's personal PACS servers. After selecting the user's PACS servers, click on the empty space in the dialog box;
     - The **Creating links using a global PACS server** option controls the restriction on creating links to studies located on a global PACS server available to all users. From the dropdown list, select the required number of global PACS servers. After selecting the global PACS servers, click on the empty space in the dialog box;
     - The **Deanonymization** option determines whether a deanonymized study link will be returned to a user sending request;
     - The **Possibility to download the study** option determines whether a download link to a study will be returned to a user sending a request;
     - The **Signing links** option controls the restriction on signing links.

4. Click the **SUBMIT** button to create the API key or **CANCEL** to cancel.

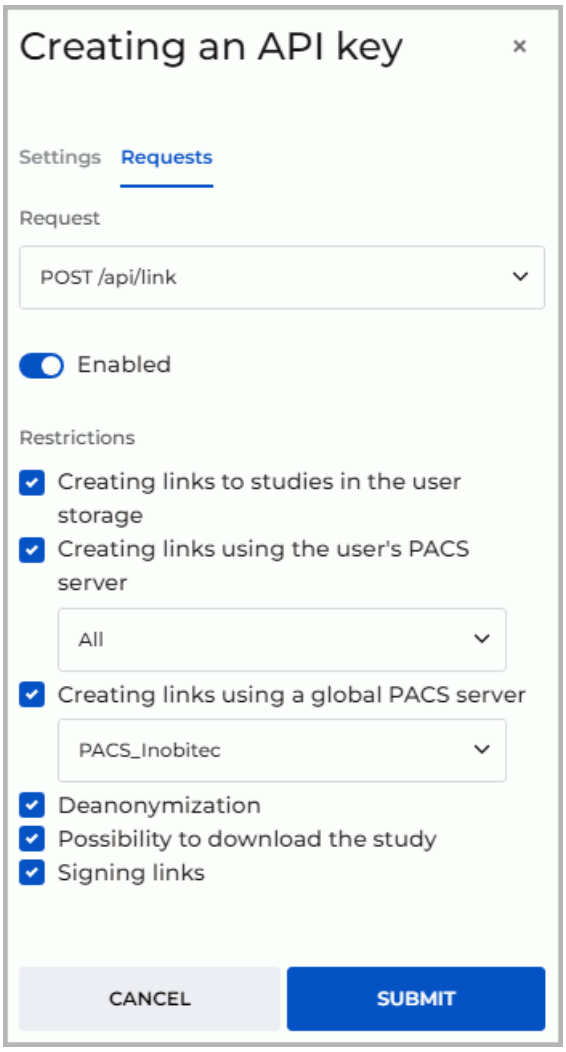


Figure 3.17: The **Requests** *tab of the* **Creating an API key** *dialog box*

To modify the parameters of an API key, do the following:

1. Select an API key from the list and edit its parameters on the **Settings** and **Requests** tabs on the right-hand side of the **API keys** page (Fig. 3.15).

2. Click the **SUBMIT** button to save the changes or **CANCEL** to cancel.

You can copy the selected key to the clipboard in two ways:

- from the read-only **Key** field on the **Settings** tab: select the text in the **Key** field, right-click on the selected text to open the context menu, and select **Copy**, or press the **Ctrl+C** key combination on the keyboard;

- from the **Key** column cell in the keys table: hover the mouse cursor over the value in the **Key** column, right-click, and select **Copy value** from the context menu.

To delete a selected API key from the list, click the **Delete API key** button  . In the confirmation dialog box that pops up, click **YES** to delete or **CANCEL** to cancel.

3.12 Links Table

On the **Links** tab (Fig. 3.18), a list of links to studies created by users is displayed. The procedure of creation links in the user's interface is described in the **Creating Links to View Studies** chapter of the User Manual.

The procedure of creating links by entering study parameters in the browser address bar and via an API request is described in Sections 5.1.1 and 5.1.2 of this Manual.

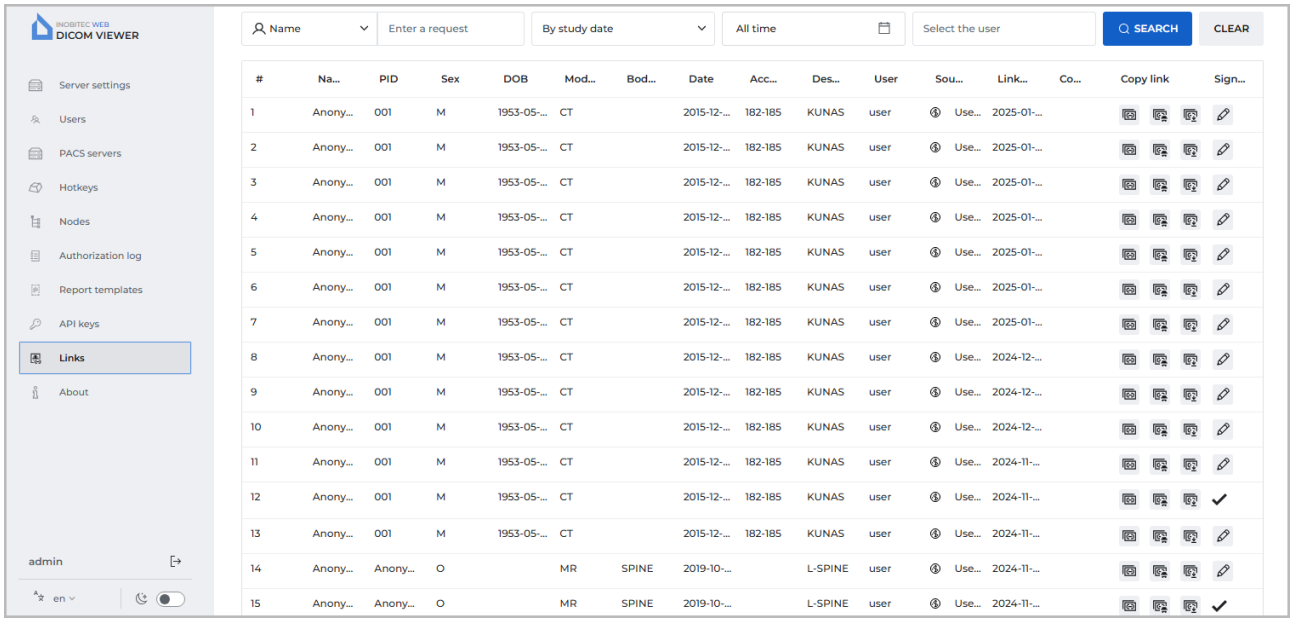


Figure 3.18: Links

For each entry on the list, information is displayed about the study and the link created for it. Study parameters are shown in the following columns:

- **#**: sequential link number;
- **Name**;
- **PID** (Patient ID);
- **Sex**;
- **DOB** (date of birth);
- **Modality**;
- **Body part**;
- **Date** of the study;

- **Accession number**;
- **Description** of the study.

Information on the link is displayed in the following columns:

- **User** (username) who created the link;
- **Source** (storage) where the study is located. If the study is hosted on a PACS server, the PACS server name is displayed in the cell;
- **Link creation date**;
- **Comment** on the link.

The administrator can perform the following actions:

- **Copy link** to the clipboard;
- **Sign link**. If a license for signing links is not available, the **Sign link** column is not displayed in the table, and the function of signing links is unavailable.

To copy a link to a study, click the **Copy link to open the study** button  in the selected line. The link is copied to the user's device clipboard.

To hide personal data when viewing a study via a link, click the **Copy link to open the anonymized study** button  . The link is copied to the user's device clipboard. When the study is opened via this link, personal data is not displayed. You cannot view structured reports, protocols, tags, and PDF documents for anonymized studies.

To copy a link that allows the user to open and download a study, click the **Copy link to open and download the study** button  . The link is copied to the user's device clipboard. When the study is opened via this link, the user gains the ability to download the study.

If a link was not signed when created, it can be signed on the **Links** page (for more details, see Sections **Creating a Link from the Study List** and **Creating a Link from the Viewing Tab** in the User Manual). To do this, click the **Sign link** button  in the corresponding line.

The server address (URL) provided in the link is the value of the *serverUrl* parameter in the **.config.json** configuration file and can be modified by the administrator. By default, the parameter value is *http://localhost:8090*. The file is located in the **html** directory.

Searching for study links is performed on the search panel similar to searching for studies. Besides, it is possible to search by the link creation date and username.

At the bottom of the **Links** page, the number of links on the current page is displayed. To go to the next page of the list, click the › button; to go to the previous page, click the ‹ button.

From the **items per page** dropdown list, select the number of links to be displayed per page. The default value is 20 links per page.

3.13 Version of the Web DICOM Viewer

To find out the Web DICOM Viewer version, click the **About** page (Fig. 3.19).

Figure 3.19: **About** *page*

The page displays the following information about the program:

- the Web DICOM Viewer version;
- Link to the website of the software developer;
- Email address of the Inobitec Software FZ-LLC Marketing and Sales Department;
- Email address of the Inobitec Software FZ-LLC Tectnical Support.

3.14 Tables

On the pages named **Users**, **PACS servers**, **Hotkeys**, **Nodes** the data are presented in tables. The user can customize the tables for better presentation of information.

3.14.1 Customizing Parameters Display Options

To customize the parameters display options, right-click on the table header or any row in the table and choose the **Table settings** option in the right-click menu. A dialog box shown in Fig. 3.20 will pop up. Our example shows the settings for the table on the **Users** page. The tables presented on the pages named **Hotkeys**, **Nodes**, **Authorization log** can be customized by analogy.

Figure 3.20: **Table settings** *dialog box*

To change the order of parameters display in the table header, move the cell with the selected parameter. To display or hide a column with a table parameter, click the **HIDE** or **SHOW** button in the cell with this parameter. Check the **Text wrapping** box to move the long value of parameters within the line. To restore the default settings, click the **BY DEFAULT** button.

To escape from the table settings dialog box, click the **CLOSE** button.

3.14.2 Sorting by Values

If there are several lines in the table, you can sort the lines by the values displayed in one of the columns. To do that, click on the header of a column with the respective parameter. Next to the name of this column, you will see an arrowhead showing the sorting order (**down** — in descending order, **up** — in ascending order). To change the sorting order, click on the column header once again. You cannot sort the lines by several parameters simultaneously.

3.14.3 Table Context Menu

To open the context menu for the table, right-click on a table row. The following commands are available on the context menu:

- **Copy value**. The value from the selected cell is copied to the clipboard;
- **Copy row**. All the *JSON* or *CSV* values are copied from the selected row to the clipboard;
- **Cancel selection** Deselects the row and the current item. On the **Users** and **Nodes** pages, the edit form is closed for the user or the node;
- **Table settings**. For details of the table settings, see Section 3.14.1.

Chapter 4

Working via a Reverse Proxy

4.1 Configuring Nginx as a Reverse Proxy

In some situations, the user needs to provide access to multiple servers in the same domain, while not using the lower level domains. For example, the user may need to place the Web DICOM Viewer at *example.com/web-viewer* address.

To perform this task, they need to use **Nginx** HTTP server configured as a reverse proxy in front of the servers. A reverse proxy works only with the servers that have been associated with it and returns responses only from these servers.

In the Web DICOM Viewer, a functionality has been implemented that allows operation via a **Nginx** reverse proxy server using:

- http connection;
- https connection with encryption support;
- redirection from http to https.

You will find the details on configuring **Nginx** as a reverse proxy at: **https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/**. The availability of the resource and relevance of the information on this link is not guaranteed. In this case, we recommend finding information on configuring the **Nginx** reverse proxy server yourself.

It is necessary to host static the Web DICOM Viewer content on the **Nginx** HTTP server for the viewer to work correctly. Simply copy the contents of the *html* directory from the installation package of the Web DICOM Viewer to the directory where **Nginx** is installed. In the example of our configuration file, the *html* directory from the Web DICOM Viewer distribution package is copied to the */html/web-viewer/* directory of the **Nginx** HTTP server.

Below you will find are examples of the **Nginx** configuration files.

4.1.1 Configuring Nginx for HTTP with static content serving and traffic proxying

Starting from version 2.10, the Web DICOM Viewer supports the functions of opening via a link and *WebSocket* proxying. Detailed documentation on setting up a tunnel between the client and the proxied server is available at: **https://nginx.org/en/docs/http/websocket.html**.

Below you can see an example of a Nginx reverse proxy server configuration file for the HTTP protocol:

```
events {
    worker_connections 1024;
}

http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    client_max_body_size 100m; # Maximum size of transferred files

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    map $http_upgrade $connection_upgrade {
        default upgrade;
        ''      close;
    }

    server {
        listen 8888; # http port

        server_name localhost; # server name

        # web-viewer access path, e.g. http://localhost:8888/web-viewer
        location /web-viewer {
            alias html/web-viewer; # path to static web-viewer content on nginx
            index index.html;
        }

        # Opening by links
        location /web-viewer/viewer/image {
            alias html/web-viewer/viewer;
            index index.html;
        }

        location /web-viewer/viewer/mpr {
            alias html/web-viewer/viewer;
            index index.html;
        }

        location /web-viewer/viewer/3d {
            alias html/web-viewer/viewer;
            index index.html;
```

```
        }

        # Proxying http requests
        # /web-viewer - web-viewer access path
        # http://192.168.0.44:8090 - server address
        location /web-viewer/v2 {
            proxy_pass http://192.168.0.44:8090/v2;
            proxy_http_version 1.1;
            proxy_set_header Connection "Keep-Alive";
            proxy_set_header Proxy-Connection "Keep-Alive";
            proxy_set_header Host $host;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Real-IP $remote_addr;
    proxy_cookie_path /v2 /web-viewer/v2;
            proxy_read_timeout 310s;
            proxy_buffering off;
            proxy_request_buffering off;
        }

        # WebSocket proxying
        # /web-viewer - web-viewer access path
        # http://192.168.0.44:8090 - server address
location /web-viewer/eventQueue {
            proxy_pass http://192.168.0.44:8090/eventQueue;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection $connection_upgrade;
        }
    }
}
```

4.1.2 Configuring Nginx for HTTP and HTTPS with static content serving and traffic proxying

Interaction with the server via HTTPS is required to support encryption in order to enhanced security. When setting up operation via HTTPS, the user must independently obtain a certificate. In Section 4.1.4 you can find an example of certificate generation.

In this example, *ssl* is enabled for port 8089, and the locations of files with the server certificate and the private key is specified. The HTTP connection is available at *http://localhost:8888/web-viewer*, while the HTTPS connection is available at *https://localhost:8889/web-viewer*. Static content for the Web DICOM Viewer is served by **Nginx**. Detailed documentation on configuring an HTTPS server is available at: **https://nginx.org/en/docs/http/configuring_https_servers.html**.

```
events {
    worker_connections 1024;
}
```

```
http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    client_max_body_size 100m; # Maximum size of transferred files

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    map $http_upgrade $connection_upgrade {
        default upgrade;
        ''      close;
    }

    server {
        listen 8888; # http port
listen 8889 ssl; # https port

        server_name localhost; # server name

ssl_certificate     ../ssl/certificate.crt; # certificate file location
        ssl_certificate_key ../ssl/privateKey.key; # location of the private key
        file

        # web-viewer access path, e.g. https://localhost:8889/web-viewer
        location /web-viewer {
            alias html/web-viewer; # path to the web-viewer static files on nginx
            index index.html;
        }

        # Opening by links
        location /web-viewer/viewer/image {
            alias html/web-viewer/viewer;
            index index.html;
        }

        location /web-viewer/viewer/mpr {
            alias html/web-viewer/viewer;
            index index.html;
        }

        location /web-viewer/viewer/3d {
            alias html/web-viewer/viewer;
            index index.html;
        }
```

```
# Proxying http requests
# /web-viewer - web-viewer access path
# http://192.168.0.44:8090 - server address
        location /web-viewer/v2 {
            proxy_pass http://192.168.0.44:8090/v2;
            proxy_http_version 1.1;
            proxy_set_header Connection "Keep-Alive";
            proxy_set_header Proxy-Connection "Keep-Alive";
            proxy_set_header Host $host;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Real-IP $remote_addr;
    proxy_cookie_path /v2 /web-viewer/v2;
            proxy_read_timeout 310s;
            proxy_buffering off;
            proxy_request_buffering off;
        }

        # WebSocket proxying
# /web-viewer - web-viewer access path
# http://192.168.0.44:8090 - server address
location /web-viewer/eventQueue {
            proxy_pass http://192.168.0.44:8090/eventQueue;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection $connection_upgrade;
        }
    }
}
```

4.1.3 Configuring Nginx for HTTP and HTTPS with proxying of all traffic

In this example, the HTTP connection is available at *http://localhost:8888*, and the HTTPS connection is available at *https://localhost:8889*. Static content for the Web DICOM Viewer is served by the Control Node.

```
events {
    worker_connections 1024;
}

http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    client_max_body_size 100m; # Maximum size of transferred files

    include /etc/nginx/mime.types;
```

```
    default_type application/octet-stream;

    map $http_upgrade $connection_upgrade {
        default upgrade;
        ''      close;
    }

    server {
        listen 8888; # http port
listen 8889 ssl; # https port

        server_name localhost; # server name

ssl_certificate     ../ssl/certificate.crt; # certificate file location
        ssl_certificate_key ../ssl/privateKey.key; # location of the private key
        file

        # Proxying http requests
        # http://192.168.0.44:8090 - server address
        location / {
            proxy_pass http://192.168.0.44:8090;
            proxy_http_version 1.1;
            proxy_set_header Connection "Keep-Alive";
            proxy_set_header Proxy-Connection "Keep-Alive";
            proxy_set_header Host $host;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_read_timeout 310s;
            proxy_buffering off;
            proxy_request_buffering off;
        }

    # WebSocket proxying
        # http://192.168.0.44:8090 - server address
location /eventQueue {
            proxy_pass http://192.168.0.44:8090/eventQueue;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection $connection_upgrade;
        }
    }
}
```

4.1.4 Example of Generating an SSL Certificate for Nginx

To enable TLS or SSL, it is necessary to create a pair comprising a public certificate and a private key. The private key is stored on the server and encrypts the data sent to clients, while the SSL certificate is available for all the users connecting connected to the server. This

provides an opportunity to decrypt the content signed with the respective SSL key. Below you will find an example of generating files for a self-signed SSL certificate and private key.

To generate a self-signed SSL certificate, OpenSSL must be installed.

To create the certificate and private key files, run the following command in the command line:

openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout privateKey.key -out certificate.crt

where,

days — is the validity period of the certificate,

keyout — is the key file name/path,

out — is the certificate file name/path.

After entering the command, a series of dialogs will pop up, prompting you to enter the required information. Once the information is provided, the files will be generated. Examples of using the generated files can be found in sections 4.1.2 and 4.1.3.

Chapter 5

The Web DICOM Viewer Integration with Third-Party Services

5.1 Creating Links to Studies

In the Web DICOM Viewer, the an ability to create links for viewing studies has been provided. Links to studies can be created:

- in the Web DICOM Viewer user interface. Information about such links is saved as entries in the table on the **Links** page of the user web interface and in the Web DICOM Viewer database. Detailed information about creating links in the user interface is provided in Chapter 8 of the User Manual **Creating Links for Viewing Studies**;
- by entering study parameters in the browser address bar. Information about links created in this way is not saved in the Web DICOM Viewer database. By default, this method of creating links is disabled. Creating links to studies by providing parameters is described in Section 5.1.1;
- via an API request (for details, see Section 5.1.2).

5.1.1 Creating Links to Studies by Providing Parameters

The function of opening studies via links created by entering study parameters in the browser address bar is disabled by default. Information about such links is not stored in the Web DICOM Viewer database. In this section, we describe how to activate the function of opening studies via links created by providing parameters.

To enable the function of opening links created by providing parameters, perform the following steps:steps:

1. Go to the directory where the Web DICOM Viewer software package is installed.
2. In the **renderserverconfig.json** configuration file (for the unified build) or **controlnodeconfig.json** configuration file (for the distributed build), change the value of the

link_by_params_policy parameter value. If there isn't such a parameter in the configuration file, it must be added.
The admissible values are:

- **forbidden** — opening links with parameters is prohibited. This is the default value;
- **anonymized** — the studies opened via links with parameters are anonymized;
- **allowed** — the studies opened via links with parameters are not anonymized;
- **download** — the studies opened via links with parameters are not anonymized and can be downloaded as archives.

3. Save the edited configuration file.
4. Restart the Web DICOM Viewer (for the unified build of the program). For the distributed build, restart the **ControlNode** service.

A link with parameters has the following format:
http(s)://<server name>/viewer/<viewing method>?<parameters>,
where:

server name — is the IP address of the server and the port on which the node is activated;

viewing method — is the viewing mode in which the series is to be opened. The available values are:

image — opening the series in the **Image viewer** window;

mpr — opening the series in the **MPR reconstruction** window;

3d — opening the series in the **Volume reconstruction** window;

parameters — stands for the query parameters of the HTTP request. These include:

– the source where the study to be opened via the link is stored. The available values are:

user=<user name> — the user's storage;

pacsServer=<PACS server name> — a PACS server shared by all the users;

user=<user name>&pacsServer=<PACS server name> — a user's personal PACS server;

– the unique identifier (UID). The available values are:

studyUid=<StudyUID> — the UID identifier of the study containing the series to be opened via the link. This corresponds to the «StudyInstanceUID» DICOM tag ;

accessionNumber=<AccessionNumber> — the Accession Number of the study. It can be specified as an identifier that is used to search for the study if the *studyUid* is not provided;

seriesUid=<SeriesUID> — the UID identifier of the series to be opened (an optional parameter). This corresponds to the «SeriesInstanceUID» DICOM tag. If the **seriesUid** parameter value is missing, the series selected by the server according to the following logic is opened:

- the first series of the study that contains images opens in the viewing mode specified in the link (image, mpr, 3d), if the series supports the selected mode;
- if the study contains no series with images, the first series opens in one of the supported modes: (**ECG viewer**, **Viewing PDF documents**, **Viewing Structured Reports**, **Watching Videos**);
- if the study contains no series with images or supported modalities, the first series of the study opens in tag viewing mode.

Below you can see some examples of links with parameters.

http://localhost:8090/viewer/image?user=<user>&studyUid=<StudyUID>&seriesUid=<SeriesUID>

When you follow this link, the series with the identifier **<seriesUid>** from the study with the identifier **<studyUid>**, which is located in the storage of the user **<user>**, is opened in the image viewer window.

http://localhost:8090/viewer/image?pacsServer=<PACS server name>&studyUid=<StudyUID>&seriesUid=<SeriesUID>

When you follow this link, the series with the identifier **<seriesUid>** from the study with the identifier **<studyUid>**, which is located on the PACS server **<PACS server name>**, is opened in the image viewer window.

5.1.2 Creating Links to Studies via API

When executing API requests, authorization in the Web DICOM Viewer is performed with API keys. API keys are created by the administrator on the **API Keys** page (see Section 3.11).

Link is created via the **POST/api/link** request. For authorization in the Web DICOM Viewer, you need to pass the API key in the request in one of the following ways:

- in the **apikey** query parameter;
- in the **X-Api-Key** header.

The request must include a *json* with the following fields:

1. **params** — object containing the link parameters:
   - **studyUid** — the UID identifier of the study containing the series to be opened via the link. This corresponds to the «StudyInstanceUID» DICOM tag;
   - **accessionNumber** — the Accession Number of the study. It is used to search for the study if the **studyUid** is not specified;
   - **seriesUid** — the UID identifier of the series to be opened via the link (an optional parameter). This corresponds to the «SeriesInstanceUID» DICOM tag. The parameter is considered if the **studyUid** is specified. If the series identifier is not specified, the series selected by the server opens. The logic for selecting the series is described in Section 5.1.1;

- **user** — username (login). This parameter is required if the study is opened from the user's storage or the user's personal PACS server;
- **pacsServer** — PACS server name. This parameter is required if the **user** parameter is not provided. If the **user** parameter is specified, the **pacsServer** parameter can be used optionally.

If the study is located in the user's storage, the **user** parameter must be provided; if it is on a global PACS server, the **pacsServer** parameter is required; if it is on the user's PACS server, both the **user** and the **pacsServer** parameters are needed.

2. **study** — an object containing the study parameters:
   - **uid** — identifier (optional parameter). If this parameter is specified while the **studyUid** or **accessionNumber** parameters are not provided in the **params** section, it is used as the **params.studyUid**. If the **studyUid** is specified in the **params** section, it must match the **uid** value;
   - **accessionNumber** — number of the order for conducting the study. If the **studyUid** or **accessionNumber** parameters are not provided in the **params** section, this value is used as the **params.accessionNumber**. If the **accessionNumber** is specified in the **params** section, the values must match;
   - **date** — date;
   - **modalities** — modality (this parameter can contain a list of values);
   - **description** — description;
   - **patient** — patient information:
     - **dob** — date of birth;
     - **id** — id;
     - **name** — name.
3. **regions** — body part (this parameter can contain a list of values);
4. **comment** — comment for the link, which is displayed in the links table (optional);
5. **sign** — flag indicating the need to sign the link.

Upon a successful response, a *json* is returned containing the following fields:

- **anonymized_link_query** — query parameters for the request to open the study in the anonymized form via a link;
- **download_link_query** — query parameters for the request to open the study via a link without anonymization, with the possibility to download;
- **link_query** — query parameters for the request to open the study via a link without anonymization;
- **signed** — information about the link signature.

Depending on the API key restrictions, some of these fields may be absent.

To form a link, you need to construct it from the server address, the opening mode, and the received parameters. The opening modes available are: **image**, **3d**, **mpr**.

The full link of the desired type is formed in compliance with the server response and has the following format:

http(s)://<server address>/viewer/<image, 3d, mpr>?<query parameters returned by the server>

Here is an example of such a link:
http://localhost:8090/viewer/image?link=eAhudwsNRPidigzHfR93Bw

Below, you can see an example of the body of a request for creating a link.

```
{
     "params": {
         "studyUid": "1.2.276.0.7230010.3.1.2.1070191241.29484.1728451815.990",
         "accessionNumber": "123456789",
         "seriesUid": "1.2.276.0.7230010.3.1.3.1070191241.29484.1728451817.1193",
         "user": "user",
         "pacsServer": "Pacs server name"
     },
     "study": {
         "uid": "1.2.276.0.7230010.3.1.2.1070191241.29484.1728451815.990",
         "accessionNumber": "123456789",
         "date": "2000-01-01",
         "description": "description",
         "modalities": ["CT"],
         "patient": {
              "dob": "1970-01-01",
              "id": "123",
              "name": "patient name"
         },
         "regions": ["HEART"]
     },
     "comment": "Example link",
     "sign": true
}
```

Below, you can see an example of the body of a response for creating a link.

```
{
    "anonymized_link_query": "link=123",
    "download_link_query": "link=123&deanonymization=456&download=789",
    "link_query": "link=123&deanonymization=456",
    "signed": true
}
```

Thanks for selecting our product! The Inobitec Software FZ-LLC is constantly working to improve it. We will be grateful for any feedback, comments and suggestions how to enhance the product functionality, user-friendliness and visualization quality.

We wish you success in your work!