

Working with Web Servers: Copying Files

Among numerous ways to configure your development and production environments the most frequent ones are as follows:

- The Web server is installed on your computer. The sources are under the server document root (for example, /htdocs), and you do your development right on the server.
- The Web server is installed on your computer but the sources are stored in another folder. You do your development, then copy the sources to the server.
- The Web server is on another computer (remote host). Files on the server are available through the FTP/SFTP/FTPS protocol, through a network share, or a mounted drive.

Now let's see how to use IntelliJ IDEA in the above environment configurations. IntelliJ IDEA assumes that all development, debugging, and testing is done on your computer and then the code is deployed to a production environment.

Please note the following:

- The reason to stick to this "local development - deployment" model lies in the way IntelliJ IDEA provides its coding assistance which includes code completion, code inspections & validations, code navigation, etc. All this functionality is based on the *index of the project files* which IntelliJ IDEA builds when the project is loaded and updates on the fly as you edit your code.
- To provide efficient coding assistance, IntelliJ IDEA needs to re-index code fast, which requires fast access to project files. The latter can be ensured only for *local* files, that is, files that are stored on your hard disk and are accessible through the file system. Therefore IntelliJ IDEA does not support the mode when you access your files over a network folder (very often it becomes slow and unresponsive, performs random look-ups for no obvious reason, etc).

Interaction between IntelliJ IDEA and servers is controlled through *server access configurations*. Anytime you are going to use a server, you need to define a *server access configurations*, no matter whether your server is on a remote host or on your computer.

Taking into account all the above, let's define the following basic concepts related to synchronization between IntelliJ IDEA and servers.

- An *in-place server* is a server whose *document root* is the parent of the project root, either immediate or not. In other words, the Web server is running on your computer, your project is under its document root, and you do your development directly on the server.
- A *local server* is a server that is running in a local or a mounted folder and whose *document root* is *NOT* the parent of the project root.
- A *remote server* is a server on another computer (remote host).
- The *server configuration root* is the highest folder in the file tree on the *local* or *remote* server accessible through the server configuration. For *in-place* servers, it is the project root.
- A *local file/folder* is any file or folder under the project root.
- A *remote file/folder* is any file or folder on the server, either local or remote.

Suppose you have a project C:/Projects/My_Project/ with a folder C:/Projects/My_Project/My_Folder and a local server with the document root in C:/xampp/htdocs. You upload the entire project tree to C:/xampp/htdocs/My_Project. In the terms of IntelliJ IDEA, the folder C:/Projects/My_Project/My_Folder is referred to as *local* and the folder C:/xampp/htdocs/My_Project/My_Folder is referred to as *remote*.

- *Upload* is copying data from the project *TO* the server, either local or remote.
- *Download* is copying data *FROM* the server to the project.

After you have configured synchronization with a server, you can upload, download, and manage files on it directly from IntelliJ IDEA. Moreover, you can suppress uploading or downloading specific files or entire folders. Finally, you can optimize your workflow by configuring content roots so specific folders are not involved in indexing, which significantly saves project indexing time.

Synchronization with servers, uploading, downloading, and managing files on them are provided via the *Remote Hosts Access* bundled plugin, which is by default enabled. If not, activate it in the **Plugins** page of the **Settings** dialog box. For details, see [Enabling and Disabling Plugins](#).

In this part:

- [Configuring Synchronization with a Web Server](#)
- [Accessing Files on Web Servers](#)
- [Uploading and Downloading Files](#)
- [Comparing Deployed Files and Folders with Their Local Versions](#)

See Also

Procedures:

- [Running](#)

Reference:

- [Deployment](#)
- [Remote Host Tool Window](#)

Web Resources:

- [Developer Community](#) 