Get Started with Eclipse Scout

Table of Contents

> Looking for something else? Visit https://eclipsescout.github.io for all Scout related documentation.

Eclipse Scout is a framework that can be used in several ways. Before you can start, you should decide which way you want to take.

Choose Wisely

First of all you need to decide what programming language you want to use for the user interface. Do you prefer Java or JavaScript?

If you choose Java, your user interface code runs on a Java server. Scout takes care of sending your UI to the browser and displaying it. It also automatically directs the user interactions to the Java server and lets you process them. This approach is called *Scout Classic*.

If you choose JavaScript, your user interface code runs in the browser. You still don't work with HTML elements directly. Instead, you use the same widget model as a Scout Java developer without needing a Java server. This approach is called *Scout JS*.

No matter what you choose, you can always adjust the existing widgets or write completely new widgets using HTML, CSS and JavaScript.

In addition to the programming language, you should think about whether you want to go full stack (front end, back end) or just focus on one tier. For example, you could use Scout JS on the front end, Scout on the back end and a REST api for the communication. Or maybe you prefer to use an existing REST api and just want to use the powerful Scout widgets for the front end. Or maybe you don't need a back end at all because you plan to implement an offline app.

Have you made your decision? Great! Then it's now time to really get started.

Installation

Depending on your decision regarding the programming language, we recommend that you start with one of the following tutorials. Both tutorials guide you through the installation process, and you will learn how to setup your first Scout Classic or Scout JS project.

Hello Scout Classic

Hello Scout JS

IDE

Even though an IDE is not required to work with Scout it is recommended to use one because it makes your developer life easier.

There are Scout plugins available for IntelliJ and Eclipse. For Scout Classic either one is fine. For Scout JS we recommend IntelliJ because the JavaScript support of Eclipse is very poor. Unfortunately, the IntelliJ Community Edition does not contain JavaScript support, so you would need IntelliJ Ultimate. There is a trial if you want to try it.

You can also use Visual Studio code or any other IDE or editor but there are no official Scout plugins for these editors.

Artifacts

All Scout artifacts are published on public repositories.

The Scout Java packages are Maven modules and available on Maven Central.

The Scout JavaScript packages are npm modules and available on npmjs.

Just include these dependencies in your project and import the required Java or JavaScript classes.

CDN

The Scout npm modules contain prebuilt scripts, stylesheets and other necessary resources in the dist folder. There are several CDNs out there that serve the content of all npm modules. A popular one is jsDelivr. Thanks to such CDNs you can easily download the prebuilt Scout assets or directly include them in your HTML files.

https://www.jsdelivr.com/package/npm/@eclipse-scout/core?path=dist

Even though this is a convenient way to get the Scout assets, we still recommend getting the artifacts via npm because it gives you more control and flexibility to work with.

If you want to learn more about using a CDN, please see Hello Scout for CDN.

Do you want to improve this document? Have a look at the sources on GitHub.