Project Information

TwoGo is a project about ride sharing and is intended for SAP employees only in the first phase. Later it shall be opened to everybody. It's sponsored by the sustainability group at SAP and focuses on our CO2 reduction goals.

The project hosts an outside-facing wiki, but relevant technical or team-internal information is stored in this team-facing wiki. Tickets are managed here as well. We used to use JIRA, switched now to Trac for its better cross-integration with source code repositories, build server and more.

Table of Contents

Development Infrastructure

The project uses Trac as project management tool with its Wiki for internal documentation, its Issue Tracker for stories, defects features and tasks, Perforce and git as VCS, JDK of version 6, Maven as build tool, Nexus to host your build artifacts and the build dependencies you require, Jenkins for continuous build and test, and Eclipse as IDE (see CodingStandard) for JAVA development, respectively WebStorm for JavaScript development. See the above links for detailed installation instructions. Developers just need to install a JDK, Perforce, Maven, and Eclipse with plugins plus the local server environment (identical with our central build and dev system). It includes the application and database server. Developers will need to install it so they can execute tests, run a full Maven build (which in turn executes the tests), run the application from Eclipse, or control the application server or inspect the database content. In order to work our Android client or our new UI/landing page (a.k.a ScriptStorm) please see additional here

Another alternative is to use a Linux VMWare Image which is prepared for the TwoGo developer. The VMWare Image contains configured Perforce Client, Eclipse IDE, Lean Java Server and MaxDB. Please see VMImage for details.

Testing is treated as integral part of the development process and infrastructure. The TwoGo team runs unit, integration, scenario and specific DSL-based tests for all developed components. In addition Findbugs and Checkstyle are used for static code analysis (limited test scope to highest priority, but for them all issues must be cleared) and Cobertura for code coverage (50 % or higher must be reached for a stable Jenkins build and 80 % for a good "sunny" build). To ensure a reasonable quality and maintainability of the code please adhere to the Code Guidelines.

A demoable version can be accessed here https://spwdfvml0898.wdf.sap.corp:8443.

The latest JavaDoc (Backend only) can be accessed on our Jenkins build

Maven project information (Backend only) can be found here

Architecture and Concepts

There is a FeatureSpec available that explains some of the remaining features in more detail.

TwoGo was shifted away from River to a mainstream technology-mix platform with higher productivity and reliability. See here for more information.

TwoGo will facilitate the services of Nokia Maps (former Navteq). Details can be found here.

TwoGos web client is implemented in SAP UI5.