

CATALIN

Top 5 Entwurfs Entscheidungen

Verwaltungssoftware Flughafen

| | |
|--|----------|
| Liste der Entscheidungen | 3 |
| Entscheidung für eine Entwicklungsumgebung | 3 |
| Entscheidung für Persistenz-Tool | 4 |
| Entscheidung für eine Versionsverwaltung | 5 |
| Entscheidung für ein Build Tool | 6 |
| Entscheidung für eine Dependency Injection Library | 7 |

Liste der Entscheidungen

Entscheidung für eine Entwicklungsumgebung

| | |
|-------------------------------------|--|
| Fragestellung | Welche Entwicklungsumgebung wird verwendet? |
| Beschreibung | |
| | |
| Datum: | 10.04.2019 |
| Entscheider, Entscheidungskreis: | Gesamtes Catalin-Team |
| Alternativen | Eclipse, IntelliJ, Netbeans, vim |
| Entscheidungs- kriterien | Usability, Integrationsmöglichkeiten in Build Systeme, Integration von VCS, Performance |
| Bewertung der Alternativen | Die Alternativen haben bekannte Probleme mit Performance und Usability oder sind allgemein langsamer. VIM fehlt es an Usability, vor allem im Bereich der Integrationen von Buildsystemen. |
| Entscheidung | Die Umsetzung erfolgt komplett mit IntelliJ |
| Begründung | IntelliJ hat als kommerzielles Produkt den besten Support. Außerdem findet man hier die besten Integrationen und die beste Performance. |
| Aspekte zur Umsetzung | Plugins: Gradle, Git, Guice, YAML... Codestyle: Standard Java |

Entscheidung für Persistenz-Tool

| | |
|-------------------------------------|--|
| Fragestellung | Wie wird Persistenz sichergestellt? |
| Beschreibung | Daten sollen auch nach Beenden des Programms gespeichert sein und bei Neustart fehlerfrei rekonstruiert werden können. |
| | |
| Datum: | 10.04.2019 |
| Entscheider, Entscheidungskreis: | Gesamtes Team |
| Alternativen | Hibernate, EclipseLink |
| Entscheidungs- kriterien | JPA muss implementiert sein, wenn möglich Integration in IDE, große Verbreitung |
| Bewertung der Alternativen | Alternativen zu unbekannt und aus Eclipse Umfeld |
| Entscheidung | Hibernate |
| Begründung | Branchenstandard, Beste Bekanntheit im Team |
| Aspekte zur Umsetzung | Standard JPA Implementierung, Guice Persistence Plugin |

Entscheidung für eine Versionsverwaltung

| | |
|-------------------------------------|---|
| Fragestellung | Welches Tool zur Versionierung wird verwendet? |
| Beschreibung | |
| | |
| Datum: | 10.04.2019 |
| Entscheider, Entscheidungskreis: | Gesamtes Catalin-Team |
| Alternativen | Git, SVN |
| Entscheidungs- kriterien | Einfaches Branching, integrierbar in Entwicklungsumgebung |
| Bewertung der Alternativen | SVN ist mittlerweile veraltet und wurde weitestgehend von Git abgelöst, Git wird bereits vom ganzen Team verwendet |
| Entscheidung | Git |
| Begründung | Beste Integration in IDE, Öffentlich auf GitHub hostbar, Sehr gute Plattform Integrationen auf GitHub |
| Aspekte zur Umsetzung | Repository wird auf GitHub gehosted, Master & Dev Branch, Vorerst keine Feature Branches |

Entscheidung für ein Build Tool

| | |
|-------------------------------------|---|
| Fragestellung | Welches Tool soll zum Bauen des Projekts benutzt werden |
| Beschreibung | |
| | |
| Datum: | 10.04.2019 |
| Entscheider, Entscheidungskreis: | Gesamtes Catalin-Team |
| Alternativen | Gradle, Maven, Ant, Bazel |
| Entscheidungs- kriterien | Schnelles Building, Gutes Caching, Breite Unterstützung von CI Plattformen, Sehr gut einstellbar |
| Bewertung der Alternativen | Alternativen wie Maven oder Ant sind entweder veraltet oder langsamer. Bazel ist noch zu unbekannt im Team, wenngleich das schnellste System, die Community aber recht klein und wenig allgemeine Akzeptanz, schlechte Unterstützung von vielen CI Plattformen. |
| Entscheidung | Gradle |
| Begründung | Gute IDE Integration, dank Daemon sehr schnell beim bauen, sehr gut einstellbar und individualisierbar |
| Aspekte zur Umsetzung | Gradle mit java und maven plugins, jar task erweitern zum Bauen von fat jar mit abhängigkeiten |

Entscheidung für eine Dependency Injection Library

| | |
|-------------------------------------|--|
| Fragestellung | Welches Tool soll zum Bauen des Projekts benutzt werden |
| Beschreibung | |
| | |
| Datum: | 10.04.2019 |
| Entscheider, Entscheidungskreis: | Gesamtes Catalin-Team |
| Alternativen | HK2, Guice, Governor, Spring |
| Entscheidungs- kriterien | Unterstützung für JSR-330, kompakte Module, Singleton Scope Unterstützung, gut testbar |
| Bewertung der Alternativen | Spring bringt ein zu großes Ökosystem mit und ist schlecht integrierbar, Governator ist von Netflix, basiert auf Guice und wird nicht mehr weiterentwickelt, HK2 wird derzeit zu Eclipse transferiert und deshalb nur begrenzt maintained |
| Entscheidung | Guice |
| Begründung | Allgemein anerkannte und einfach zu verwendende Library, Weiterentwicklung durch Google, Gute Integration in IDE |
| Aspekte zur Umsetzung | JSR Annotations statt Guice Annotations verwenden, Module für verschiedene Persistenz Modi, JPA mit Guice JPA Extension |