

SPECCHIO Web Interface

Software Architektur Dokument

Datum: 31.12.2016
Remo Rossi, Christian Schibli

Version	Datum	Name	Kommentar
0.1	30.10.16	Christian Schibli	Erstellung und Grobstruktur
0.2	04.11.16	Christian Schibli	Deployment-Prozess
0.3	31.12.16	Christian Schibli	Entwicklungsumgebung
0.4	03.12.16	Christian Schibli	Projekt in Eclipse aufsetzen
0.5	06.12.16	Christian Schibli	Weiterentwicklung
0.6	30.12.16	Christian Schibli	Dokumentstruktur anpassen, Problembehebung, Zugangsdaten, Kurzübersicht SPECCHIO Web Interface, Systemkontext, Technologien
0.7	07.01.17	Christian Schibli	Grobarchitektur, Feinarchitektur
0.8	16.01.17	Remo Rossi	Revidieren
1.0	18.01.17	Christian Schibli	Letzte Korrekturen, Layout, finale Version

Inhaltsverzeichnis

1	Ziel des Software Architekt Dokuments (SAD)	5
2	Dokument Scope	5
3	Kurzübersicht SPECCHIO Web Interface	6
3.1	Search	6
3.2	Search Result.....	7
3.3	Detail View.....	8
3.4	About	9
3.5	Contact.....	10
4	Glossar	11
5	System-Kontext.....	12
6	Technologien.....	13
6.1	Übersicht.....	13
6.2	Google Maps API.....	14
7	Grobarchitektur.....	15
8	Feinarchitektur.....	16
8.1	Ablaufdiagramm	16
8.2	Klassendiagramm	17
8.3	Java Resources	18
8.3.1	ch.specchio.controller	18
8.3.2	ch.specchio.model.....	18
8.3.3	ch.specchio.util.....	18
8.4	WebContent.....	18
8.4.1	css	18
8.4.2	fonts	19
8.4.3	img.....	19
8.4.4	js	19
8.4.5	WEB-INF.....	19
8.4.6	about.jsp.....	19
8.4.7	contact.jsp	19
8.4.8	detail.jsp	19
8.4.9	error.jsp	20
8.4.10	footer.html	20
8.4.11	head.html	20
8.4.12	nav.html	20
8.4.13	search.jsp.....	20
8.4.14	searchResult.jsp.....	20
9	Entwicklungsumgebung.....	21
9.1	Aufbau	21
9.2	Glassfish-Server.....	22
9.3	SPECCHIO Virtual Machine	23
9.4	Git Repository	24

9.5	Projekt in Eclipse aufsetzen	26
9.5.1	Glassfish Tools	26
9.5.2	Glassfish Server in Eclipse erstellen.....	29
9.5.3	Projekt importieren	33
9.5.4	Projekt starten.....	39
9.5.5	Probleme beim Starten des Projekts.....	41
10	Deployment.....	42
10.1	WAR-Export	42
10.2	Deployment-Prozess	43
10.3	Speicherort für SPECCHIO-Zugangsdaten (db_config.txt)	47
11	Zugangsdaten	47
12	Referenzen.....	47

1 Ziel des Software Architekt Dokuments (SAD)

Das SAD beantwortet Fragen über den System-Kontext und deren Limitationen, wie die Komponenten der Software zusammenarbeiten und wie sie implementiert worden sind. Es gibt Aufschluss über die verwendeten Technologien und deren Konfiguration. Man findet Beschreibungen der Architektur nach Top-Down Verfahren. Das heisst von einem allgemeinen Level bis ins Detail. Nach Fertigstellung des Software Architektur Dokumentes dient es als „Developer Guide“ für zukünftige Implementationen. Aus diesem Grund wurde auch ein besonderes Augenmerk auf die Entwicklungsumgebung gesetzt (Kapitel 9), damit das Setting für eine reibungslose Weiterentwicklung gewährleistet ist.

2 Dokument Scope

Beim vorliegenden Dokument handelt es sich um die Beschreibung des SPECCHIO Web Interface. Für Fragen zu den anderen SPECCHIO-Komponenten wie SPECCHIO-Server mit Datenbank oder SPECCHIO Client App, wird auf die detaillierten Dokumentations-Unterlagen auf www.specchio.ch verwiesen.

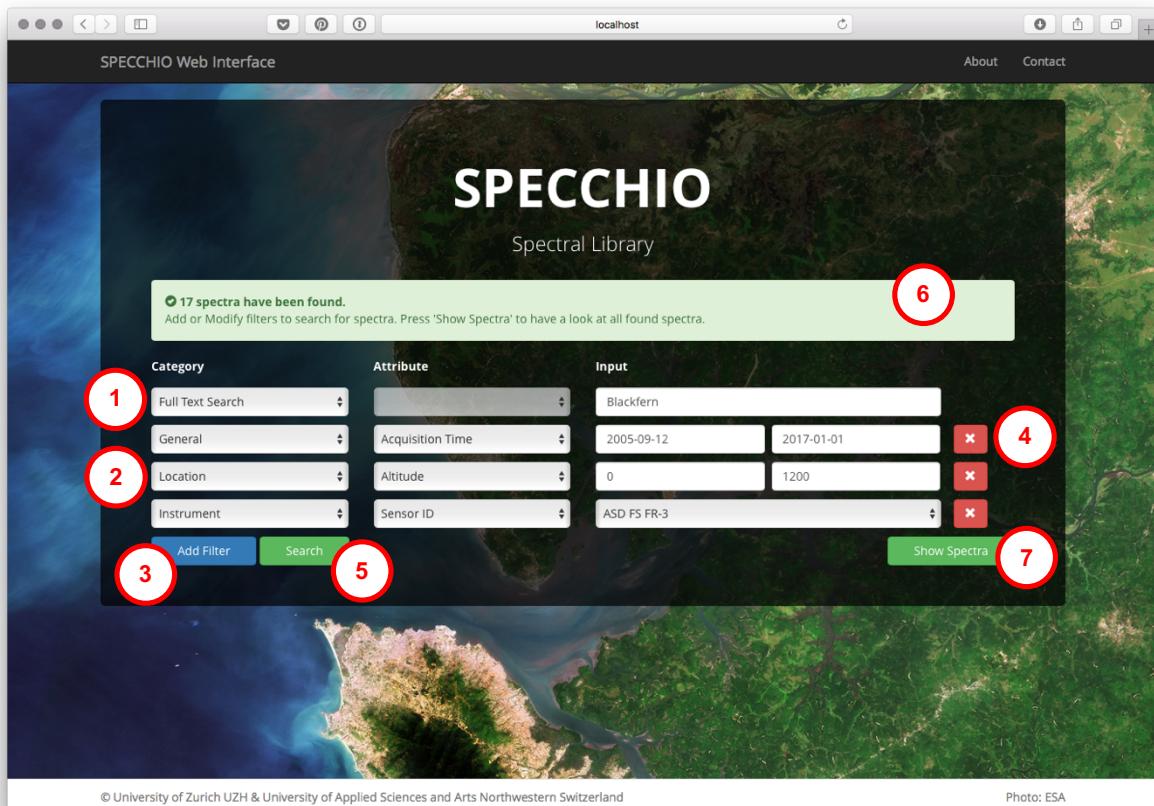
Für den Betrieb und Unterhalt (z.B. Backups) des Web-Servers auf dem die SPECCHIO Web Interface Applikation läuft, ist der Auftraggeber selber verantwortlich.

3 Kurzübersicht SPECCHIO Web Interface

Beim SPECCHIO Web Interface handelt es sich um einen Webbrowser-basierten Zugang zur SPECCHIO Datenbank. Bislang war dies nur mit spezieller Software möglich. Um eine Übersicht zum Funktionsumfang des Web Interface zu erhalten, wird nachfolgend mit Screenshots die Applikation präsentiert.

3.1 Search

Die Search Seite ist der Einstiegspunkt des SPECCHIO Web Interfaces. Hier kann durch benutzerdefinierte Filter nach Spektren gesucht werden.



1. Volltext-Suche: ermöglicht die Suche nach Teilwörtern eines Datenbankwertes.
2. Nach Kategorie und entsprechenden Attributen filtern, analog SPECCHIO Client Applikation.
3. Weitere Filter-Zeile hinzufügen.
4. Filter-Zeile löschen.
5. Suche auf SPECCHIO-Datenbank auslösen.
6. Feedback Panel: Zeigt die Anzahl gefundener Spektren oder allfällige Validierungsfehler an.
7. Weiterleitung zu der Search Result Seite.

3.2 Search Result

Auf der Search Result Seite werden die Suchergebnisse der Search Seite in einer Liste und auf einer Karte angezeigt.

The screenshot shows the SPECCHIO Web Interface with the following numbered annotations:

- 1**: Shows the number of search results per page (10).
- 2**: Sorts the results by acquisition time.
- 3**: Provides a search bar.
- 4**: Shows the map interface.
- 5**: Shows the search results table.
- 6**: Shows the "Show Detail" button.
- 7**: Shows the search result map.

Search Result Table

	Acquisition Time	Campaign Name	Investigator	Name	File Name	Institute
1	2005-09-12 10:09:50	SPECCHIO Tutorial	SPECCHIO	Blackfern	bfern.001	
2	2005-09-12 10:09:54	SPECCHIO Tutorial	SPECCHIO	Blackfern	bfern.002	
3	2005-09-12 10:09:58	SPECCHIO Tutorial	SPECCHIO	Blackfern	bfern.003	
4	2005-09-12 10:10:06	SPECCHIO Tutorial	SPECCHIO	Blackfern	bfern.004	
5	2005-09-12 10:10:18	SPECCHIO Tutorial	SPECCHIO	Blackfern	bfern.005	
6	2005-09-12 10:10:23	SPECCHIO Tutorial	SPECCHIO	Blackfern	bfern.006	
7	2005-09-12 10:10:28	SPECCHIO Tutorial	SPECCHIO	Blackfern	bfern.007	
8	2005-09-12 10:10:32	SPECCHIO Tutorial	SPECCHIO	Blackfern	bfern.008	
9	2005-09-12 10:10:37	SPECCHIO Tutorial	SPECCHIO	Blackfern	bfern.009	
10	2005-09-12 10:10:42	SPECCHIO Tutorial	SPECCHIO	Blackfern	bfern.010	

Showing 1 to 10 of 17 entries

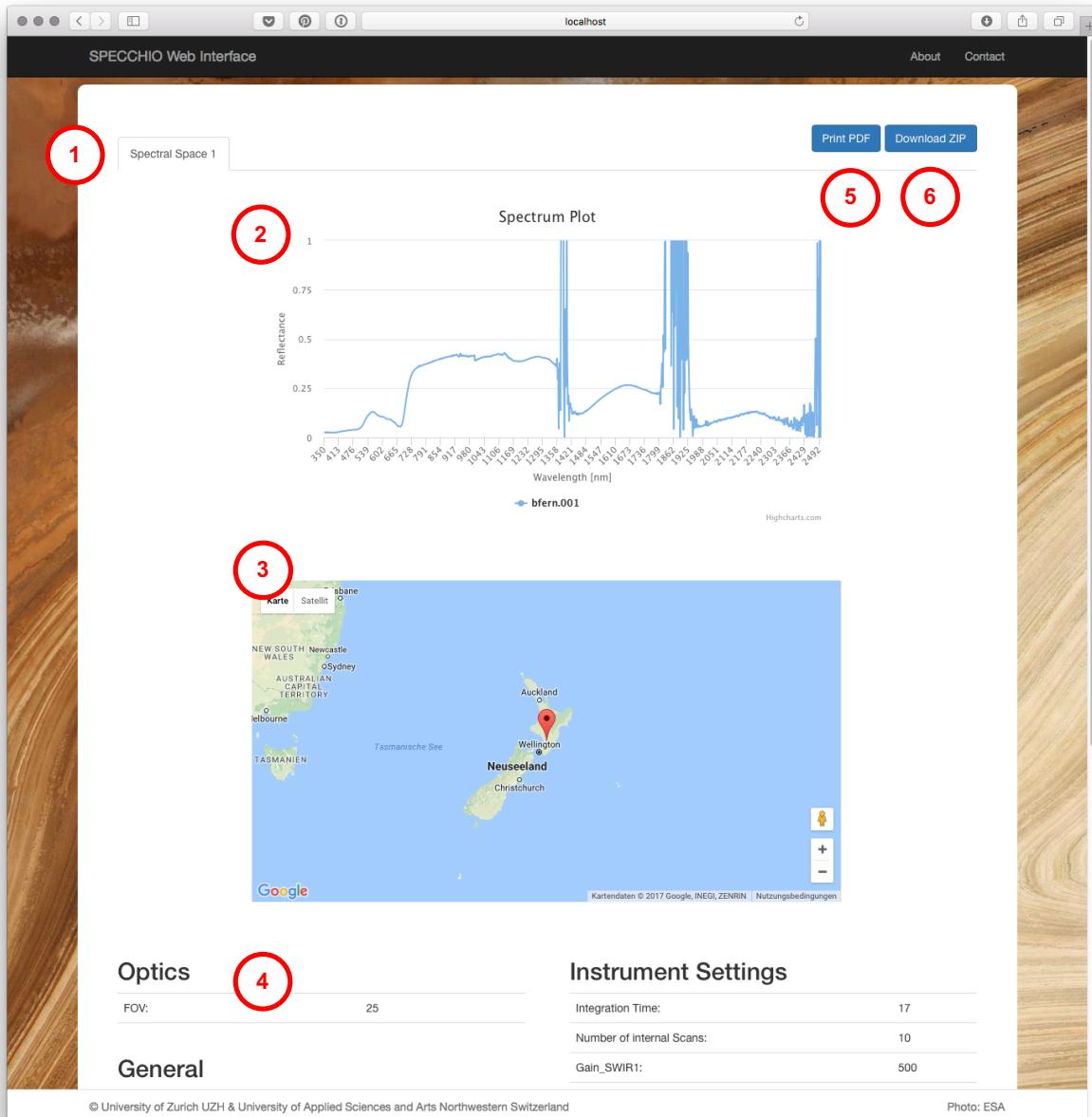
Search Result Map

The map shows several red markers indicating the locations of the search results. A tooltip is visible over one marker, showing the file name. The map includes labels for countries like Australia, New Zealand, and South America, and oceans like the Indian, Pacific, and Southern Oceans.

1. Anzahl Suchergebnisse pro Seite anzeigen.
2. Nach gewünschter Spalte sortieren, auf- oder absteigend.
3. Innerhalb der Suchergebnisse nach bestimmten Datensätzen suchen.
4. Paginierung
5. Gewünschte Datensätze auswählen.
6. Ausgewählte Datensätze anzeigen.
7. Google Map: Datensätze, die über Längen- und Breitengrade verfügen, werden in dieser Map als Marker angezeigt. Beim MouseOver eines Markers öffnet sich ein Tooltip, der den File Name des Datensatzes anzeigt. Beim Klick auf den Marker wird der entsprechende Datensatz angezeigt.

3.3 Detail View

Die Detail View enthält alle Informationen eines Spektrums oder mehrerer Spektren.



1. Da gewisse Spektren nicht miteinander verglichen werden können, gibt es eine Unterteilung in verschiedene Spectral Spaces. Pro Spectral Space wird hier ein Tab erstellt, in welchem alle Informationen der Spektren dieses Spaces enthalten sind.
2. Linien-Diagramm: Bei MouseOver werden Spektralwerte und File Name als Tooltip angezeigt.
3. Google Map mit Marker aller Spektren, die über Längen- und Breitengrade verfügen.
4. Anzeige der erfassten Metadaten. Leere Attribute und Kategorien werden nicht angezeigt.
5. PDF Export des aktuellen Tabs.
6. Export einer ZIP-Datei, welche die Spektraldaten aller Tabs als CSV-Datei enthält.

3.4 About

Die About Seite bietet einen kurzen Überblick über die SPECCHIO Web Interface und dessen involvierte Institutionen.

The screenshot shows the 'About' page of the SPECCHIO Web Interface. The background is a satellite image of Earth's surface. At the top, there is a dark header bar with the text 'SPECCHIO Web Interface' on the left and 'localhost' in the center. On the right side of the header are 'About' and 'Contact' links. Below the header, the word 'About' is prominently displayed in large white letters, followed by the subtitle 'Project and institutions'. A section titled 'Project' contains a detailed paragraph about the development of the application in cooperation with the University of Applied Sciences and Arts Northwestern Switzerland. Below this paragraph is a 'More' link. Another section titled 'Institutions' lists two partners: the University of Zurich and the University of Applied Sciences and Arts Northwestern Switzerland. Each partner entry includes its logo, name, department, and a 'More' link. At the bottom of the page, there is a copyright notice and a photo credit.

SPECCHIO Web Interface

About Contact

About

Project and institutions

Project

The Department of Geography of the University of Zurich maintains a database for spectral data and their metadata. Access to the data of this application was to date only available with a specific software. The SPECCHIO Web Interface has been developed in cooperation with the University of Applied Sciences and Arts Northwestern Switzerland to ensure that the database can be searched directly for spectral data by web browser without installing additional software.

More

Institutions

University of Zurich
Department of Geography
More

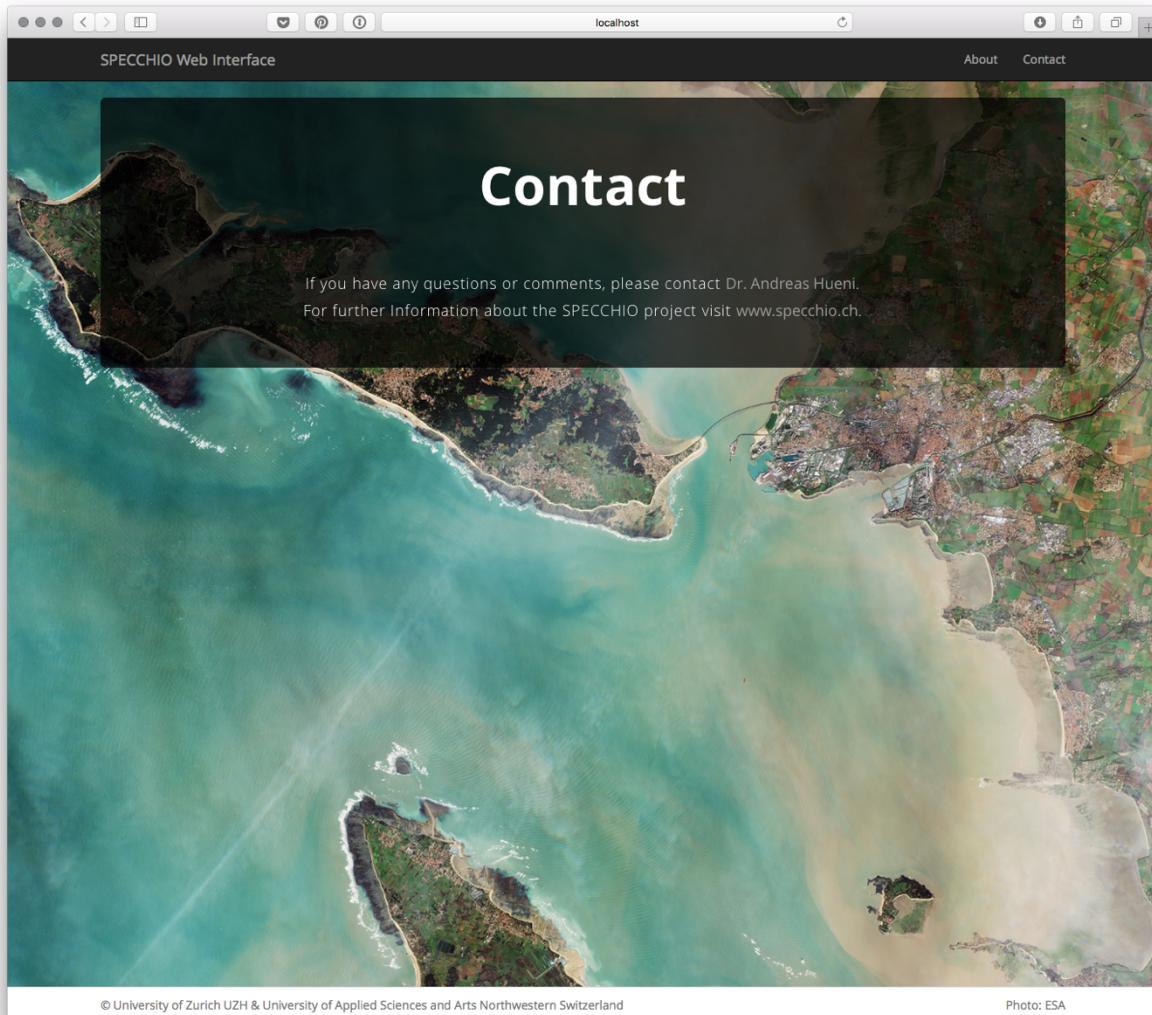
University of Applied Sciences and Arts Northwestern Switzerland
School of Engineering
More

© University of Zurich UZH & University of Applied Sciences and Arts Northwestern Switzerland

Photo: ESA

3.5 Contact

Die Contact Seite enthält die wichtigsten Kontaktinformationen.

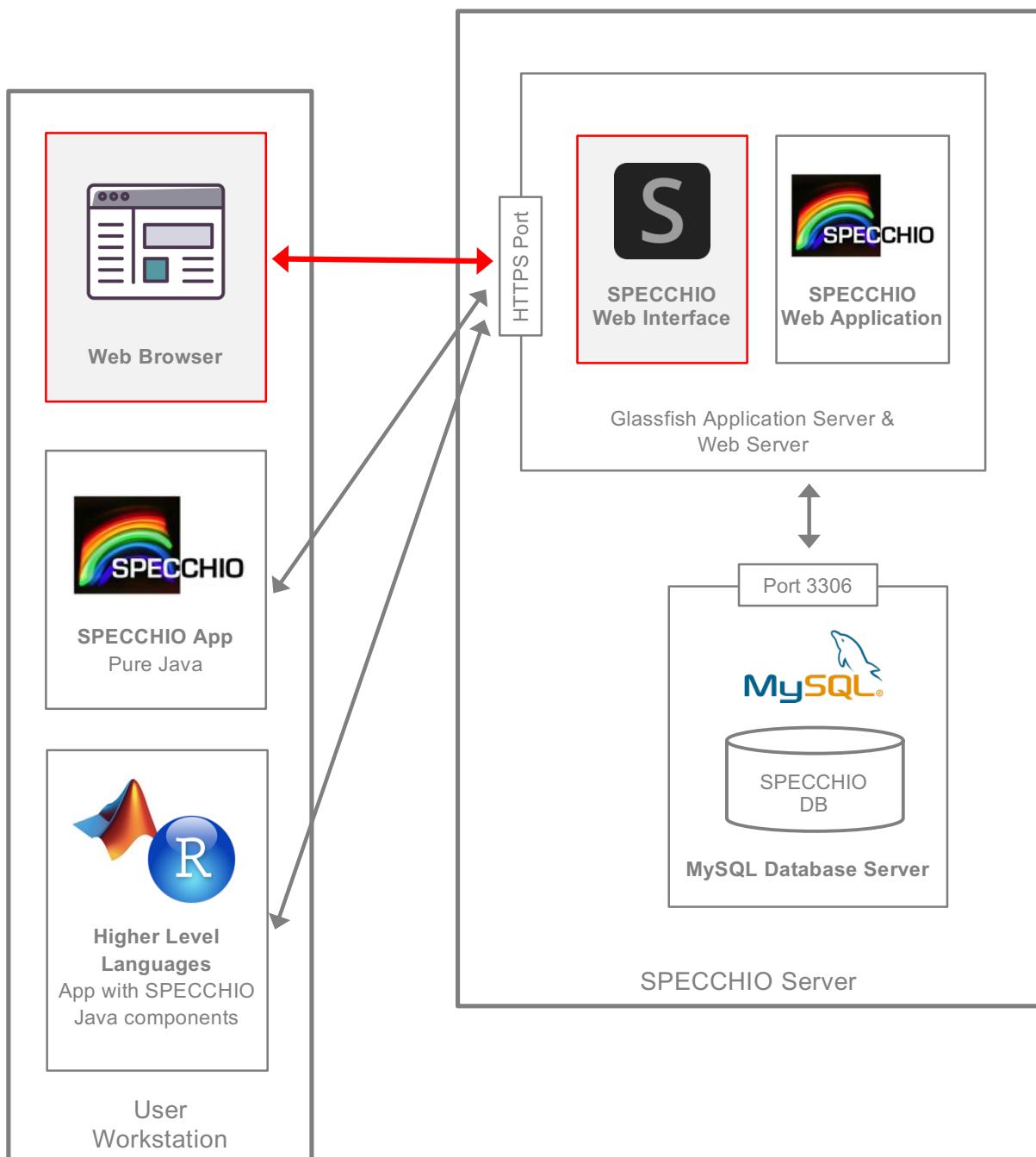


4 Glossar

Begriff	Erklärung
Git	Zentrale Ablageverwaltung unserer Projektdaten und des Source Code.
Glassfish-Administrations-Konsole	Dient zur Konfiguration des Glassfish-Servers. Erreichbar über http://localhost:4848 (Glassfish-Server muss in Eclipse gestartet sein).
Google Developers Console	Verwaltung der Google APIs auf ein bestimmtes Google-Konto lautend. Beim SPECCHIO Web Interface handelt es sich um die Google Maps API.
MVC	Model View Controller
IDE	Integrated Development Environment -> hier im Projekt: Eclipse
SPECCHIO	Ist ein Informationssystem zur Aufnahme von Referenzspektren und spektralen Kampagnendaten, die durch Spektroradiometer erfasst werden. Die Einbindung von umfangreichen Metadaten-Sets sorgt für die dauerhafte Haltung von Spektraldataen und ermöglicht die gemeinsame Nutzung von Spektraldataen zwischen Forschungsgruppen. Auf die Daten von SPECCHIO kann zur Zeit nur mit einem spezialisierten Programm zugegriffen werden.
SPECCHIO Web Interface	Ermöglicht den direkten Zugriff auf die SPECCHIO-Datenbank ohne Installation zusätzlicher Software. Es sind nur Abfragen möglich, es können keine Daten editiert werden.
VirtualBox	Software zur Virtualisierung von Betriebssystemen.
WAR	Ist ein Dateiformat und steht für Web Application Archive. Dieses Format beschreibt wie eine vollständige Webanwendung nach der Java-Servlet-Spezifikation in eine Datei im JAR- bzw. ZIP-Format verpackt wird (www.wikipedia.de)

5 System-Kontext

Bestehend waren die grau umrandeten Komponenten. Es war bislang nur mit spezifischer Software möglich, Daten der SPECCHIO-Datenbank zu lesen und zu ändern. Rot umrandet sind die Komponenten des SPECCHIO Web Interface gekennzeichnet. Mit dem SPECCHIO Web Interface ist der Lesezugriff auf die SPECCHIO-Datenbank via Webbrower möglich.



6 Technologien

6.1 Übersicht

Technologie	Verwendungszweck	Version
SPECCHIO API	Zugriff auf Datenbank und erstmalige Aufarbeitung der Daten	3.2.1.6-Alpha
Java	Webapplikation-Backend. Aufarbeitung der Daten aus der SPECCHIO API fürs Frontend	7
JSP	Einbettung von Java Code in HTML-Frontend	2.2
Java Servlets	Request/Response Handling	3.0
Gson	Umwandlung von Java- in JSON-Objekte und zurück	2.8.0
JavaScript	Dynamische Erzeugung von GUI Elementen und Bereitstellung von GUI Funktionalität	Browser-spezifisch
JQuery	Dynamische Erzeugung von GUI Elementen und Bereitstellung von GUI Funktionalität	2.1.4
Bootstrap	CSS Framework zur Darstellung des User Interfaces	3.3.7
Highcharts.com	Darstellung der Liniendiagramme	latest
Google Maps API	Darstellung der Landkarten	latest
HTML	Strukturierung des Frontends	HTML5
Oracle Glassfish	Webserver	3.1

6.2 Google Maps API

Für die Darstellung der Landkarten wird die Google Maps API verwendet. Hierzu ist ein Authentifizierungs Key notwendig, welcher über die Google Developers Console verwaltet wird. Der Zugang zur Google Developers Console mit dem entsprechenden Google-Konto ist in einem separaten Dokument notiert (siehe Kapitel 11).

The screenshot shows the Google APIs console interface. On the left, the sidebar has 'API Manager' selected, with 'Dashboard' currently active. The main area displays the 'SPECCHIO Web Interface' project, which is checked. Below it, there are links for 'My Project' and 'Mehr Projekte aufrufen'. To the right, there are three cards: 'Traffic' (Anfragen/s), 'Fehler' (Prozentualer Anteil der Anfragen), and 'Medianlatenz' (Millisekunden). Below these cards is a table showing API usage statistics:

API	Anfragen	Fehler	Fehlerverhältnis	Latenz, Median	Latenz, 98 %
Google Maps Directions API	-	-	-	-	-
Google Maps Distance Matrix API	-	-	-	-	-
Google Maps Elevation API	-	-	-	-	-
Google Maps Geocoding API	-	-	-	-	-
Google Maps JavaScript API	-	-	-	-	-
Google Places API Web Service	-	-	-	-	-

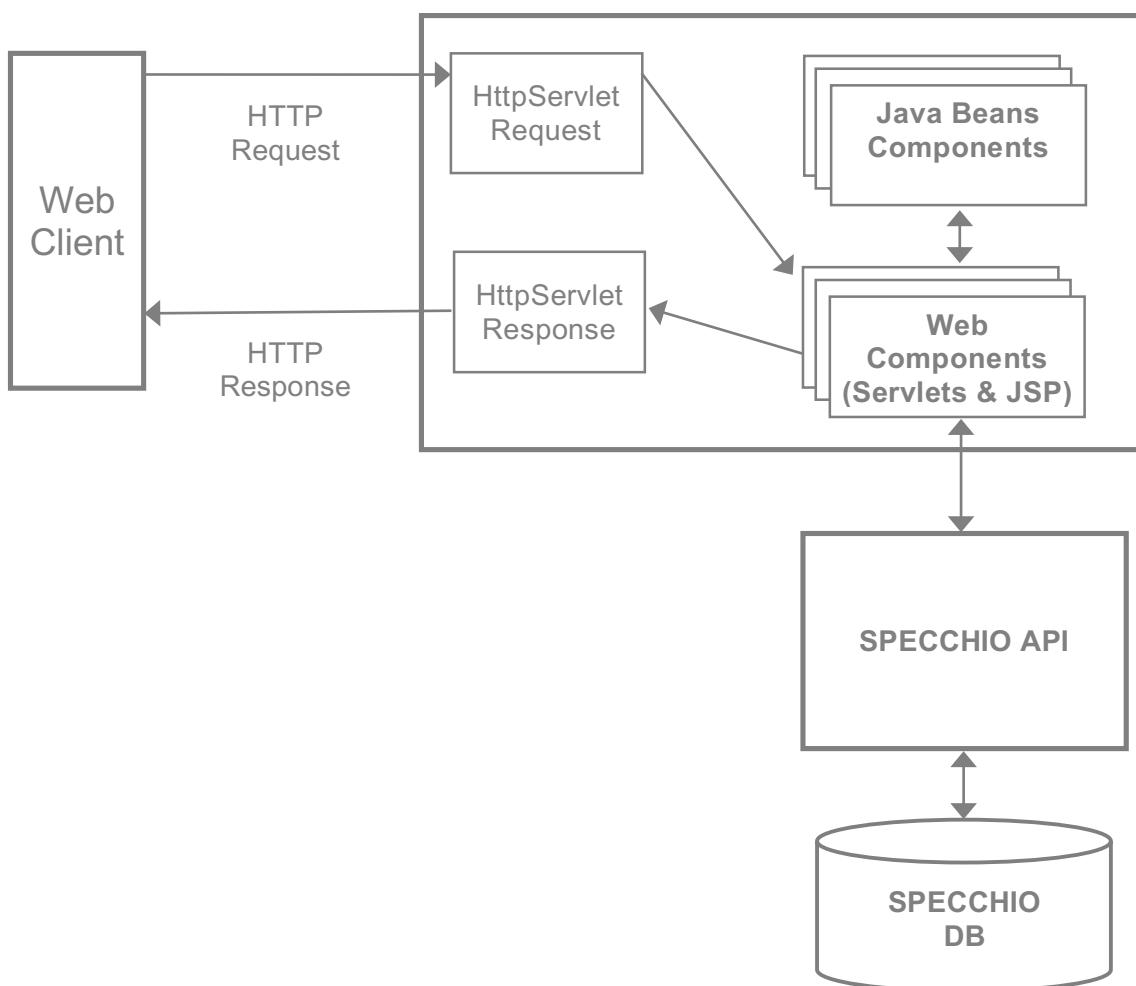
The screenshot shows the 'Zugangsdaten' tab in the Google APIs console. The sidebar still shows 'API Manager' and 'Zugangsdaten' is highlighted with a red circle. The main area has tabs for 'Anmeldedaten', 'OAuth-Zustimmungsbildschirm', and 'Domainbestätigung'. Below these tabs is a button 'Anmeldedaten erstellen' and a 'Löschen' button. A note says: 'Erstellen Sie Anmeldedaten, um auf Ihre aktivierten APIs zuzugreifen. Weitere Informationen finden Sie in der API-Dokumentation.' Below this is a table for 'API-Schlüssel':

Name	Erstellungsdatum	Einschränkung	Schlüssel
specchio	29.12.2016	Keine	A1zaSyDxVi6lkT-8txoP3nRgNeyfoJbhR-z74

7 Grobarchitektur

Das SPECCHIO Web Interface baut in der Basis auf einem MVC-Modell auf. Das Backend (Controller und Model) sind mit Java implementiert und verwenden die bestehende API, welche die Interaktion mit der Datenbank ermöglicht. Das Frontend (Views) wurde mit JSP, Javascript/JQuery und CSS implementiert. Außerdem wurden diverse Libraries und Frameworks wie Bootstrap, Google Map API und Highcharts verwendet.

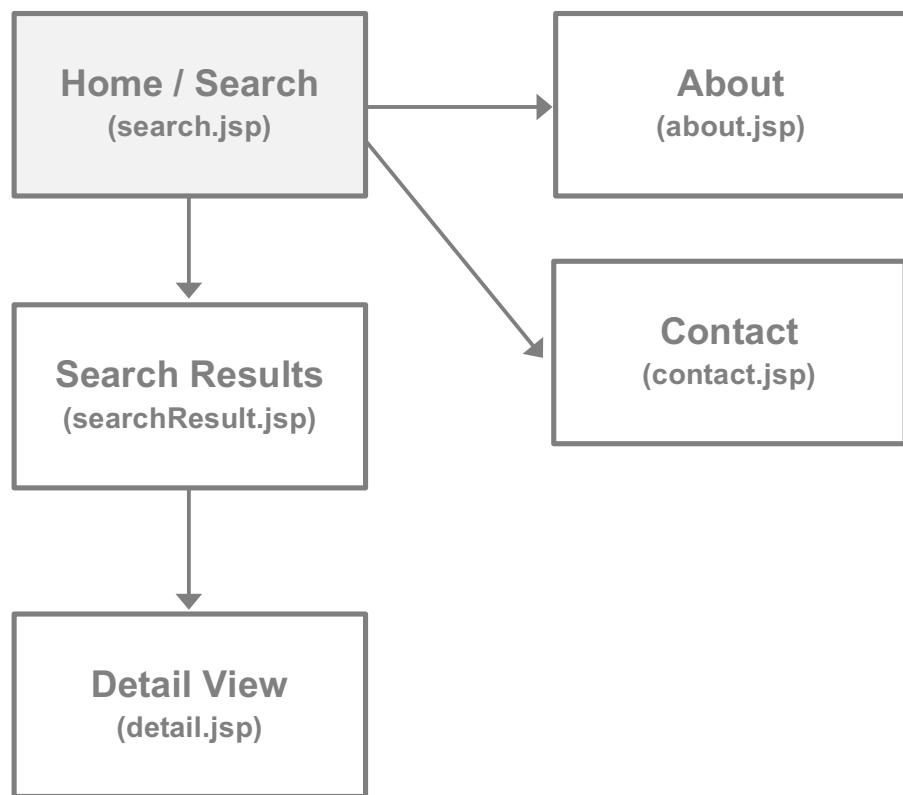
Die untere Grafik zeigt den Ablauf zwischen Client und SPECCHIO-DB auf (oracle.com/javaee):



8 Feinarchitektur

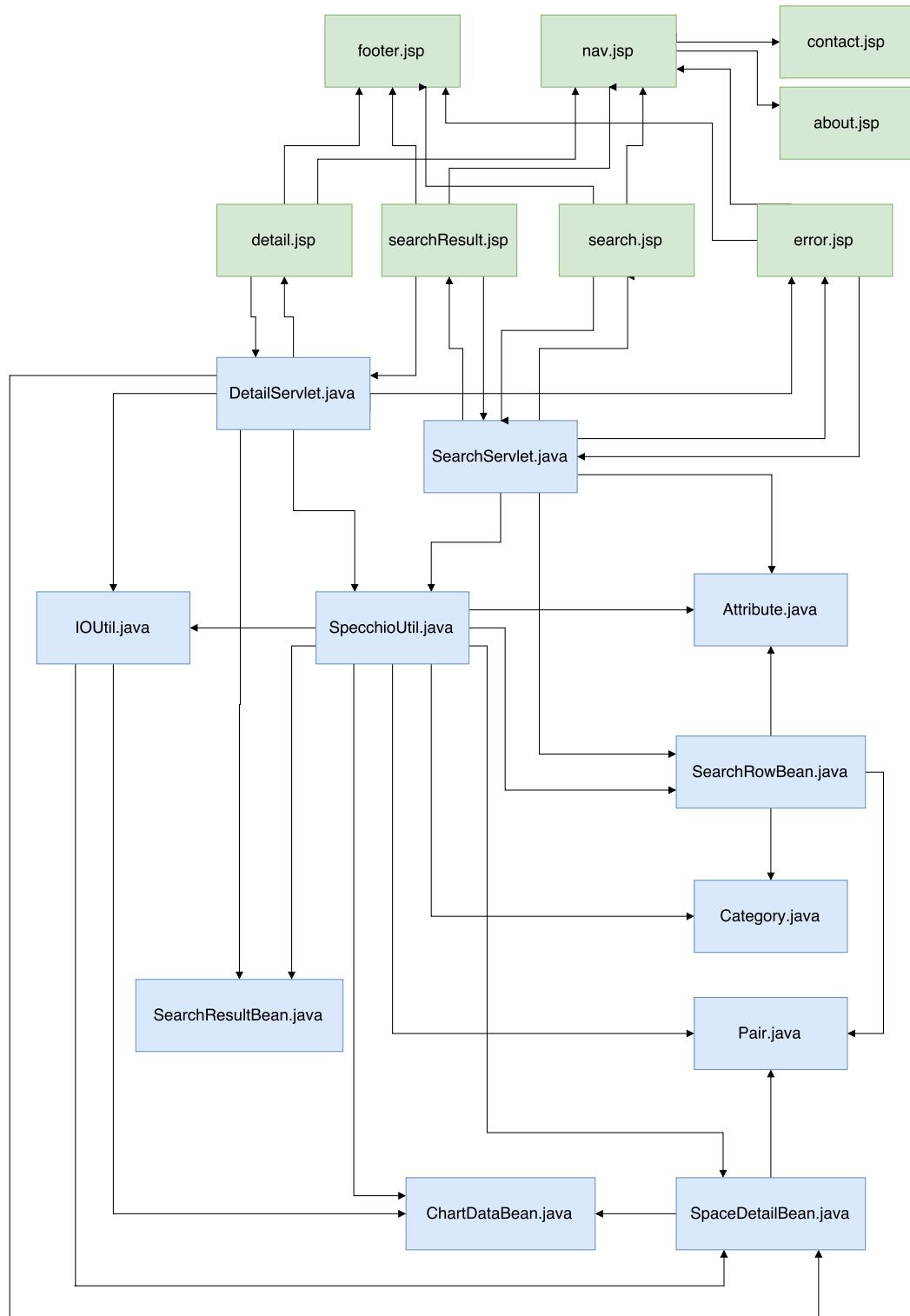
In diesem Kapitel werden die einzelnen Klassen des SPECCHIO Web Interface kurz erklärt. Weitere Informationen zu einzelnen Methoden sind direkt im Source Code kommentiert.

8.1 Ablaufdiagramm



8.2 Klassendiagramm

Das untenstehende Diagramm enthält eine Übersicht über alle Java-Klassen (blau) und JSPs (grün) des SPECCHIO Web Interfaces und dessen Verbindungen untereinander.



JavaScript- und CSS-Files, welche von den JPS importiert werden, wurden übersichtlichkeitshalber weggelassen.

8.3 Java Resources

8.3.1 ch.specchio.controller

DetailServlet.java	Wickelt die Http-Request- und Response-Prozesse für die Detail View ab. Interaktion mit der SPECCHIO API.
SearchServlet.java	Wickelt die Http-Request- und Response-Prozesse für die Such und Such Resultat View ab. Interaktion mit der SPECCHIO API.

8.3.2 ch.specchio.model

Attribute.java	Model für ein Attribute-Objekt der Metadaten.
Category.java	Model für ein Category-Objekt der Metadaten.
ChartDataBean.java	Enthält Name und Werte für das Liniendiagramm in der Detail View.
Pair.java	Ein polymorphes Objektpaar.
SearchResultBean.java	Enthält die Informationen für die Suchergebnis-Liste (searchResult.jsp). Hierzu gehören die Daten für die Suchergebnis-Liste, Koordinaten für die Marker auf der Google Map sowie die Spektrum-ID.
SearchRowBean.java	Enthält die die vom Benutzer eingegeben Inputs einer Suchzeile auf der Startseite (search.jsp).
SpaceDetailBean.java	Enthält die Informationen für einen einzelnen Tab in der Detail View (detail.jsp). Hierzu gehören Space Typ, Measurement Unit, Diagramm-Daten, Koordinaten für Google Map und alle Metadaten für jedes Spektrum.

8.3.3 ch.specchio.util

IOUtil.java	Enthält alle Methoden, bei welchen Files eingelesen oder ausgegeben werden (z.B. CSV-Export, db_config.txt).
SpecchioUtil.java	Enthält Methoden zur Interaktion mit der SPECCHIO API, Aufarbeitung der Daten und weitere Helfermethoden für die Servlets.

8.4 WebContent

8.4.1 css

Grundsätzlich kam für das Design das CSS-Framework Bootstrap zum Einsatz. Trotzdem waren viele individuelle Anpassungen nötig, so dass für jedes JSP- und HTML-File eine separate CSS-Datei im Verzeichnis „css“ angelegt wurde.

Weitere Informationen zu Bootstrap: <http://getbootstrap.com/>.

8.4.2 fonts

Im Verzeichnis „fonts“ sind lediglich die Glyph-Icons abgelegt, welche im Tabellen-Header der Suchergebnis-Liste.

Die Grundschrift „Open Sans“ ist via Google API eingebunden und ist im HTML-File „head.html“ abgelegt.

8.4.3 img

Im Verzeichnis „img“ sind alle Bildressourcen abgelegt. Hierzu gehören die Hintergrundbilder der ESA (siehe Referenzen), Logos von Universität Zürich und Fachhochschule Nordwestschweiz sowie die Favicons.

8.4.4 js

Für die drei Ansichten „search“, „searchResult“ und „detail“ existiert je eine einzelne JavaScript-Datei:

- search.js
- searchResult.js
- detail.js

Zu Bootstrap gehören:

- bootstrap.min.js
- ie10-viewport-bug-workaround.js

Für die Darstellung der Suchergebnisse:

- dataTables.bootstrap.min.js
- jquery.dataTables.min.js

jQuery:

- jquery-2.1.4.min.js

8.4.5 WEB-INF

Enthält XML-Daten zur Konfiguration und einen Ordner mit den nötigen Jar-Files.

8.4.6 about.jsp

Beinhaltet HTML-Konstrukt für die About-Site.

8.4.7 contact.jsp

Beinhaltet HTML-Konstrukt für die Contact-Site.

8.4.8 detail.jsp

Beinhaltet HTML-Konstrukt für die Detail View. Hinzu kommen hier die Einbettung des Liniendiagramms und der Google Map:

- Liniendiagramm: highcharts.com
- Google Map: Hierzu muss ein Key hinterlegt sein. Das dazugehörige Google-Konto für die Google API ist in den Zugangsdaten abgelegt (siehe Kapitel 0).

8.4.9 error.jsp

Diese Site dient zum Abfangen von Fehlermeldungen.

8.4.10 footer.html

Ist Bestandteil aller JSP-Dateien und beinhaltet den Footer.

8.4.11 head.html

Ist Bestandteil aller JSP-Dateien. Beinhaltet Meta-Tags, Einbettung des Favicons, Einbettung von Bootstrap und Einbettung der Grundschrift Open Sans.

8.4.12 nav.html

Ist Bestandteil aller JSP-Dateien und beinhaltet die Navigation.

8.4.13 search.jsp

Beinhaltet HTML-Konstrukt für die Such-Site resp. Startseite.

8.4.14 searchResult.jsp

Beinhaltet HTML-Konstrukt für die Suchergebnisse-Site. Hinzu kommen hier die Einbettung der DataTable und der Google Map:

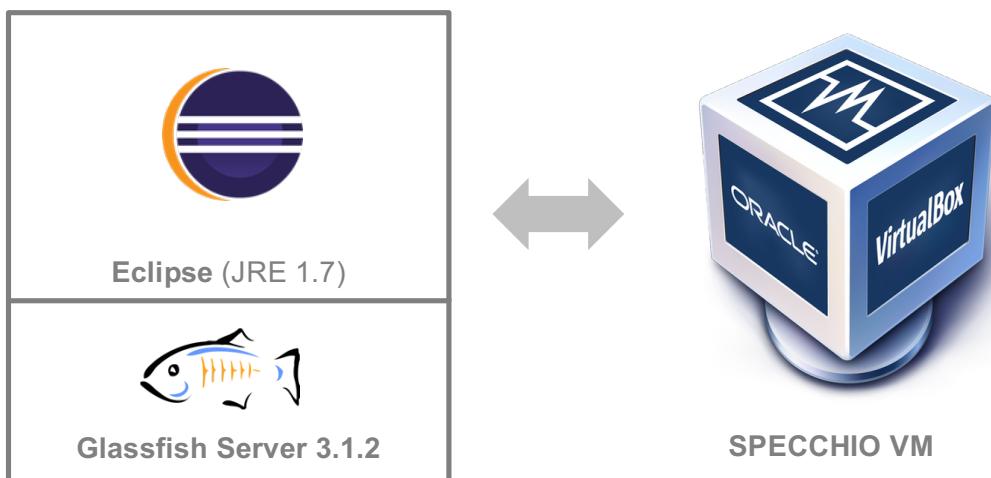
- DataTable: `dataTables.bootstrap.min.css`
- Google Map: Hierzu muss ein Key hinterlegt sein. Das dazugehörige Google-Konto für die Google API ist in den Zugangsdaten abgelegt (siehe Kapitel 0).

9 Entwicklungsumgebung

Das SPECCHIO Web Interface wurde mit Eclipse erstellt. In den nachfolgenden Kapiteln werden die Schritte zum Aufsetzen des Eclipse-Projekts beschrieben, um das Projekt mittels einem Glassfish-Server über die SPECCHIO Virtual Machine laufen zu lassen.

9.1 Aufbau

In der untenstehenden Grafik wird der Aufbau der Entwicklungsumgebung schematisch dargestellt. Alle drei Komponenten (IDE, Server, VM) werden lokal auf dem Computer installiert.



Die IDE (Eclipse) enthält das Projekt und ist mit dem Glassfish-Server verknüpft.

Die Virtual Machine enthält eine komplette SPECCHIO-Umgebung (Client, Server, Datenbank).

9.2 Glassfish-Server

Da die SPECCHIO Spectral Library mit einem Glassfish-Server 3 betrieben wird, wurde für das SPECCHIO Web Interface ebenfalls ein Glassfish-Server 3 gewählt. Dieser kann von dieser Website heruntergeladen werden:

<https://glassfish.java.net/download-archive.html>

GlassFish Server Open Source Edition 4.1 (Java EE 7)			
Platform	Java EE 7	Locale	
Zip (Win, Linux, Mac, Solaris)	Web Profile Full Platform	EN EN	glassfish-4.1-web.zip glassfish-4.1.zip

GlassFish Server Open Source Edition 4.0 (Java EE 7)			
Platform	Java EE 7	Locale	
Windows	Web Profile Full Platform	EN EN	glassfish-4.0-web-windows.exe glassfish-4.0-windows.exe
Windows	Web Profile Full Platform	ML ML	glassfish-4.0-web-windows-ml.exe glassfish-4.0-windows-ml.exe
Linux, Mac, Solaris	Web Profile Full Platform	EN EN	glassfish-4.0-unix-ml.sh glassfish-4.0-unix.sh
Linux, Mac, Solaris	Web Profile Full Platform	ML ML	glassfish-4.0-unix-ml.sh glassfish-4.0-unix.ml
Zip (Win, Linux, Mac, Solaris)	Web Profile Full Platform	EN EN	glassfish-4.0-web.zip glassfish-4.0.zip
Zip (Win, Linux, Mac, Solaris)	Web Profile Full Platform	ML ML	glassfish-4.0-web-ml.zip glassfish-4.0-ml.zip

GlassFish Server Open Source Edition 3.1.2.2 (Java EE 6)			
Platform	Java EE 6	Locale	
Windows	Web Profile Full Platform	EN EN	glassfish-3.1.2.2-web-windows.exe glassfish-3.1.2.2-windows.exe
Windows	Web Profile Full Platform	ML ML	glassfish-3.1.2.2-web-windows-ml.exe glassfish-3.1.2.2-windows-ml.exe
Linux, Mac, Solaris	Web Profile Full Platform	EN EN	glassfish-3.1.2.2-web-unix.sh glassfish-3.1.2.2-unix.sh
Linux, Mac, Solaris	Web Profile Full Platform	ML ML	glassfish-3.1.2.2-web-unix-ml.sh glassfish-3.1.2.2-unix-ml
Zip (Win, Linux, Mac, Solaris)	Web Profile Full Platform	EN EN	glassfish-3.1.2.2-web.zip glassfish-3.1.2.2.zip
Zip (Win, Linux, Mac, Solaris)	Web Profile Full Platform	ML ML	glassfish-3.1.2.2-web-ml.zip glassfish-3.1.2.2-ml.zip
AIX	Web Profile Full Platform	EN EN	glassfish-3.1.2.2-web-aix.sh glassfish-3.1.2.2-web-aix.zip glassfish-3.1.2.2-aix.sh glassfish-3.1.2.2-aix.zip

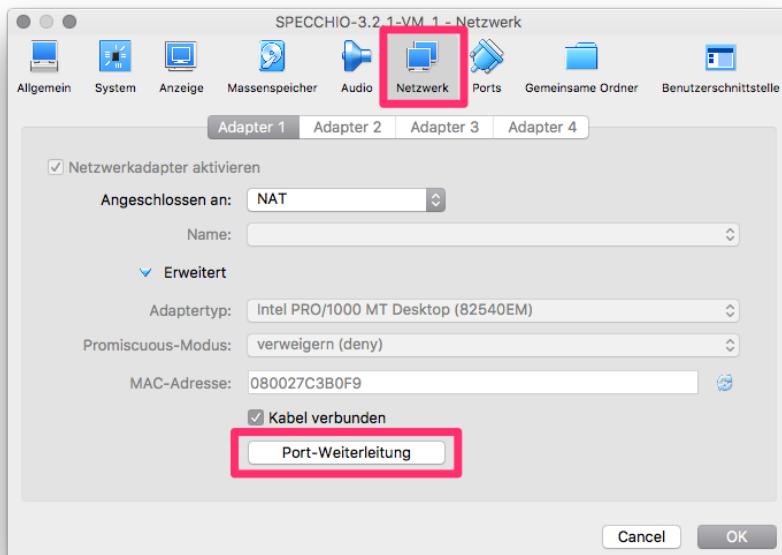
1. Download ZIP-Package (Win, Linux, Mac, Solaris)
2. ZIP-Package entpacken → darin ist ein Ordner mit der Bezeichnung "glassfish3" enthalten.
3. Den Ordner "glassfish3" an beliebige Stelle auf dem Computer verschieben.
4. Weitere Schritte mit dem Glassfish-Server werden in Kapitel 9.5 beschrieben.

9.3 SPECCHIO Virtual Machine

Die SPECCHIO Virtualbox enthält eine komplette SPECCHIO-Server- und -Client-Installation und läuft auf einem virtuellen CentOS-Betriebssystem mit Oracle Virtualbox. Eine Anleitung zur Installation der SPECCHIO Virtualbox ist zusammen mit den notwendigen Download-Links auf der offiziellen SPECCHIO-Website zu finden:

http://specchio.ch/vm_download.php

Ist die SPECCHIO Virtualbox einmal aufgesetzt, sind unter den Einstellungen der SPECCHIO VM im Netzwerk-Tab die nachfolgenden Einstellungen für die Port-Weiterleitung wichtig:



Name	Protokoll	Host-IP	Host-Port	Gast-IP	Gast-Port	Actions
HTTP	TCP	127.0.0.1	8080	10.0.2.15	8080	
HTTP External	TCP	130.60.16.221	8080	10.0.2.15	8080	
HTTPS	TCP	127.0.0.1	8181	10.0.2.15	8181	
HTTPS External	TCP	130.60.16.221	8181	10.0.2.15	8181	

Diese Einstellungen sind relevant für den Zugriff von ausserhalb der VM.

9.4 Git Repository

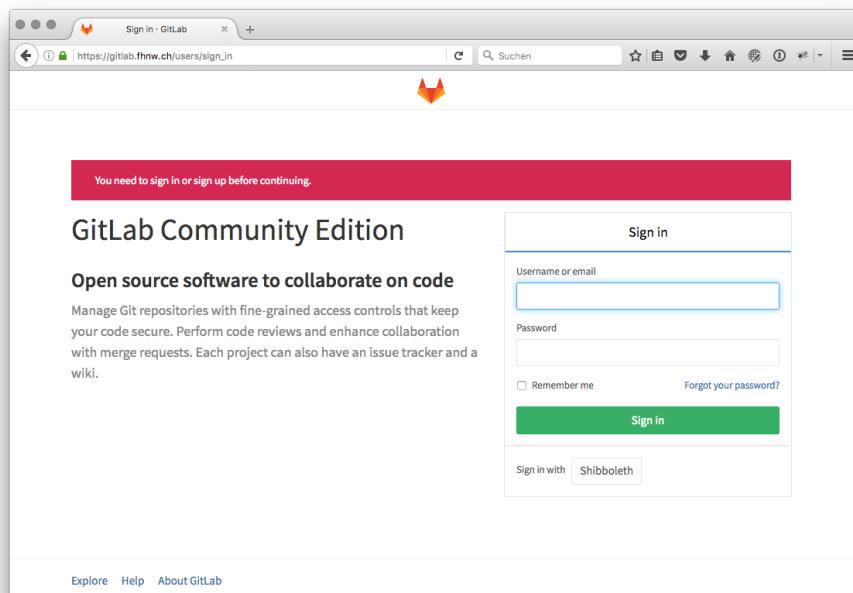
Der Quellcode befindet sich auf einem Git Repository der FHNW. Für den Zugriff auf das Repository benötigt man entsprechende Rechte, die einem vom Owner des Repositories (martin.gwerder@fhnw.ch) zur Verfügung gestellt werden müssen.

Die URL um sich mit dem Repository zu verknüpfen lautet:

https://gitlab.fhnw.ch/IP56/SPECCHIO_Webinterface.git

Das Repository kann über einen entsprechenden Gitlab-Account online eingesehen werden. Für einen Antrag zur Benutzung des Repositories wende man sich ebenfalls an den Repository-Owner (martin.gwerder@fhnw.ch).

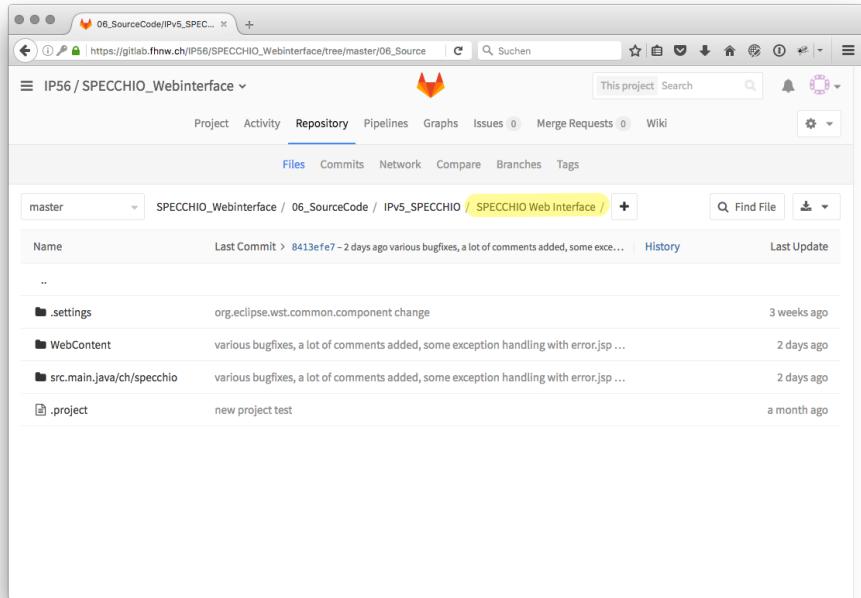
Die URL zum Login lautet https://gitlab.fhnw.ch/users/sign_in.



The screenshot shows the GitLab interface for the 'SPECCHIO_Webinterface' repository. At the top, there's a header bar with project navigation (Project, Activity, Repository, Pipelines, Graphs, Issues, Merge Requests, Wiki) and a search bar. Below the header is a banner stating 'Signed in successfully.' A large circular profile picture of a person with the letter 'S' is displayed. The main content area shows the repository details: 'SPECCHIO_Webinterface' (forked from 'IP56'), 0 stars, 0 forks, and a link to the HTTPS version of the repository. Below this, there are buttons for 'Add Changelog', 'Add License', 'Add Contribution guide', and 'Set Up CI'. A commit history section shows a single commit: '8413ebe7 various bugfixes, a lot of comments added, some exception handling with error.jsp call ...' by 'Remo Rossi' 2 days ago. A note below states: 'This project does not have a README yet' and provides instructions to add one.

The screenshot shows the 'Files' tab of the 'SPECCHIO_Webinterface' repository. The 'Repository' tab is selected. The page lists files in the 'master' branch. The files are:

Name	Last Commit	History	Last Update
01_Admin	Add minutes by 3. November 2016, mockups, deployment instructions, updated review...	8413ebe7	2 days ago
02_Meetings	Add minutes of client meeting from 16.12.2016.		a week ago
03_Arbeitsjournal	Add logfile from 31.12.2016.		3 days ago
04_Specification	Update Planning, update requirements, add change request document, add Release Pl...		a month ago
05_Planning	Update Planning, update requirements, add change request document, add Release Pl...		a month ago
06_SourceCode	various bugfixes, a lot of comments added, some exception handling with error.jsp call ...		2 days ago
07_Concepts	Add design and drafts.		a month ago
08_Documentation	Add minutes by 3. November 2016, mockups, deployment instructions, updated review...		a month ago
09_Releases	warfile 1.0		2 weeks ago
Test_Readme.md	next test (using alternate ssh access)		3 months ago



Das Eclipse-Projekt heisst “SPECCHIO Web Interface” und ist im Git Repository im Verzeichnis “06_SourceCode / IPv5_SPECCHIO /“ zu finden.

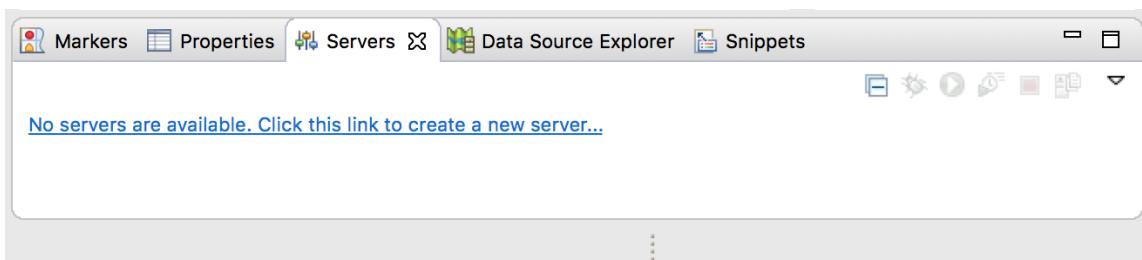
9.5 Projekt in Eclipse aufsetzen

Um das Eclipse-Projekt aufzusetzen werden die Kapitel 9.2, 9.3, 9.4 vorausgesetzt. Ausserdem ist die Installation von Eclipse for Java EE Developers notwendig, da ein Server integriert werden können muss. Da das SPECCHIO Web Interface aus Kompatibilitätsgründen mit dem SPECCHIO-Server mit einem Glassfish-Server 3.1 betrieben wird, muss zudem JDK 1.7 installiert sein.

9.5.1 Glassfish Tools

Falls noch kein Glassfish-Server in Eclipse installiert wurde, müssen zuerst die Glassfish Tools installiert werden:

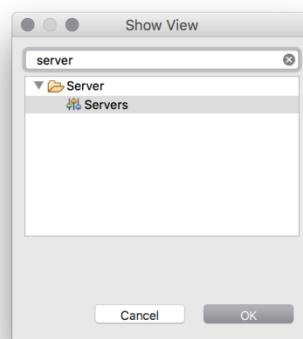
1. Servers-View öffnen.



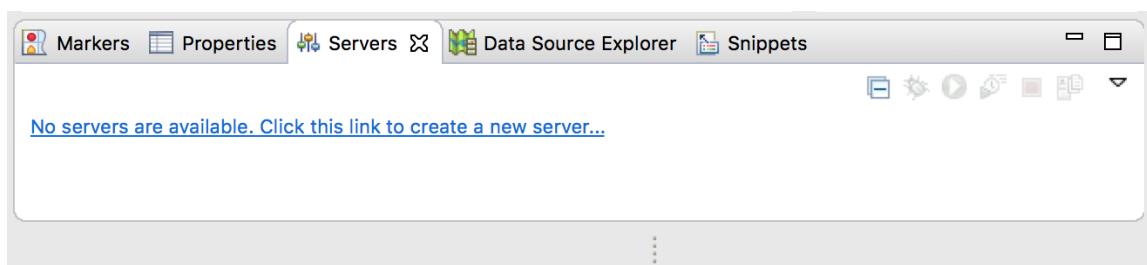
Falls die Servers-View noch nicht geöffnet ist, muss das folgende Menü geöffnet werden:

- a. Window → Show View → Other...
- b. Dann im Suchfeld “Server” eintippen

- c. Die Kategorie Server enthält die Servers-View. "Servers" auswählen und mit "OK" bestätigen.

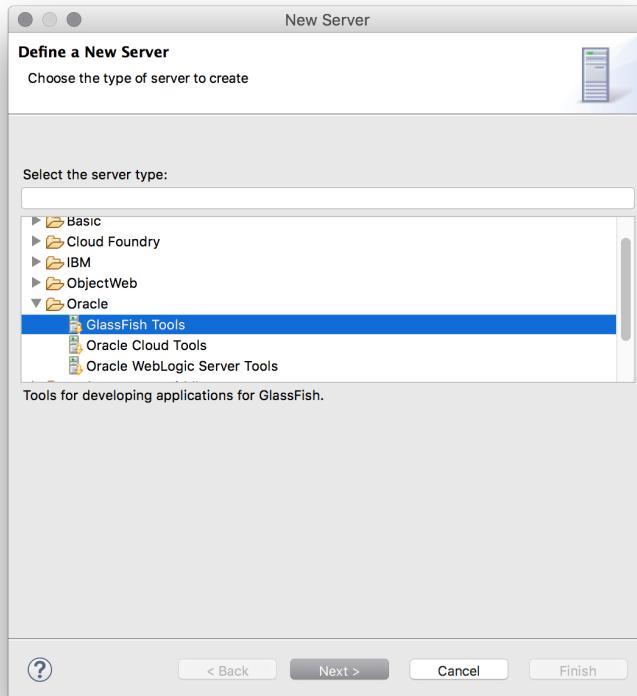


2. GlassFish Tools installieren.



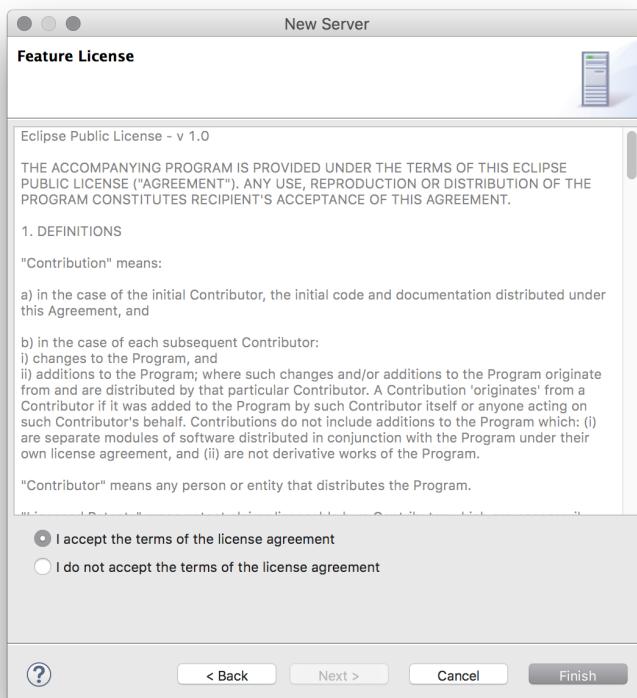
In der Servers-View die blau unterstrichene Textzeile "*No servers are available. Click this link to create a new server...*" anklicken.

3. Server-Typ wählen.

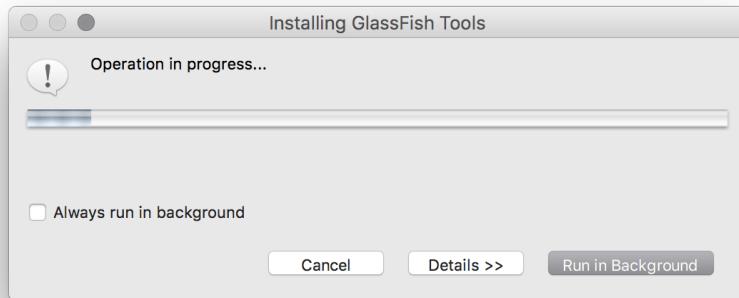


Im Verzeichnis "Oracle" die "GlassFish Tools" auswählen und mit "Next" bestätigen.

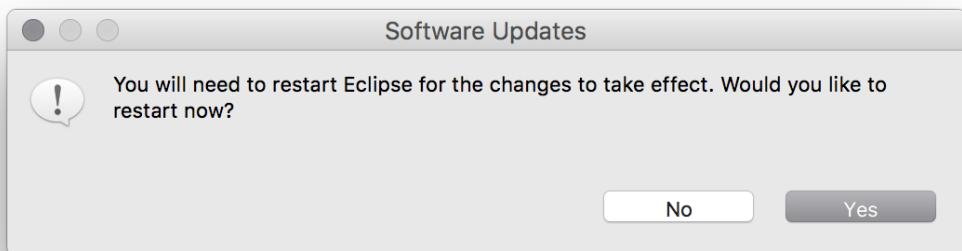
4. Lizenz akzeptieren und mit "Finish" abschliessen.



5. Die Installation der GlassFish Tools beginnt. Dies kann einen Moment dauern.



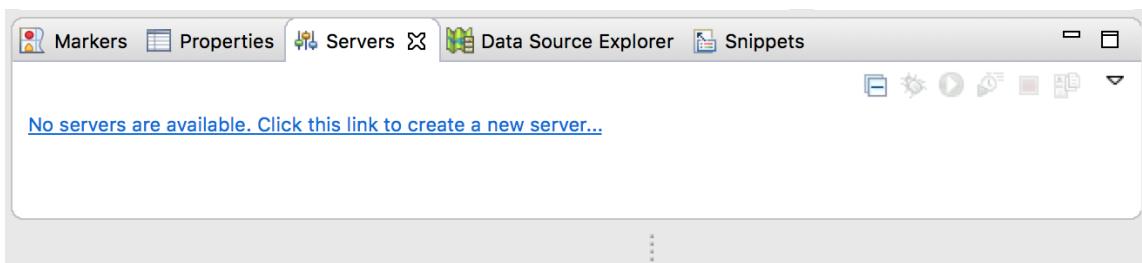
6. Nach einem Neustart ist Eclipse vorbereitet, um einen neuen Glassfish-Server zu erstellen (siehe nachfolgendes Kapitel 9.5.2)



9.5.2 Glassfish Server in Eclipse erstellen

Voraussetzung für dieses Kapitel ist die Installation der GlassFish Tools (siehe Kapitel 9.5.1).

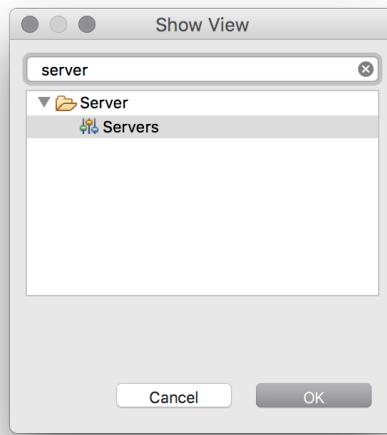
1. Servers-View öffnen.



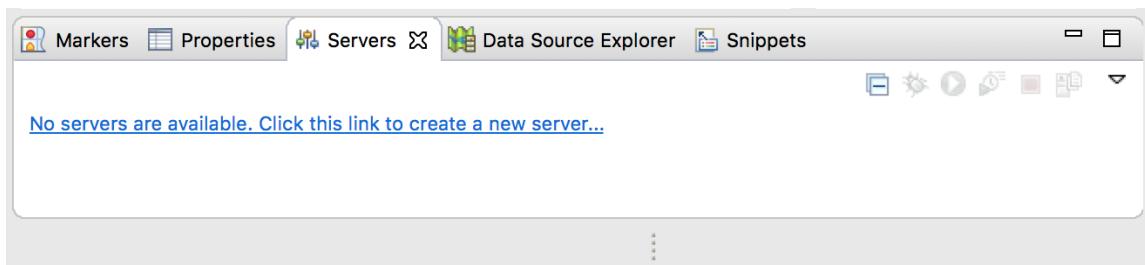
Falls die Servers-View noch nicht geöffnet ist, muss das folgende Menü geöffnet werden:

- a. Window → Show View → Other...
- b. Dann im Suchfeld "Server" eintippen

- c. Die Kategorie Server enthält die Servers-View. "Servers" auswählen und mit "OK" bestätigen.

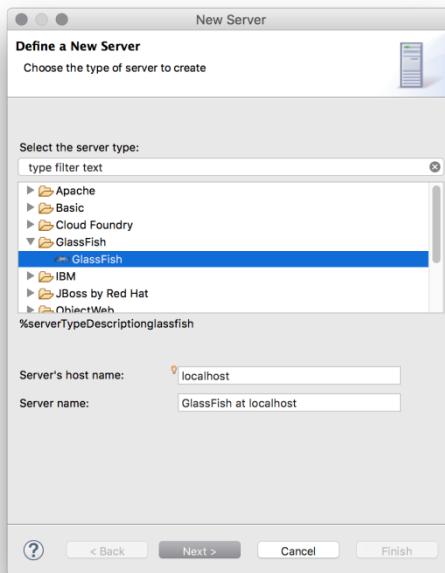


2. Neuen Server erstellen.



In der Servers-View die blau unterstrichene Textzeile "*No servers are available. Click this link to create a new server...*" anklicken.

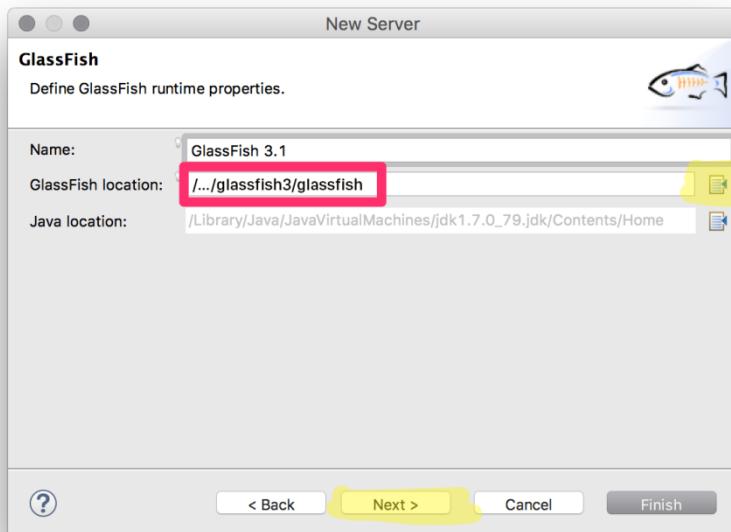
3. Server-Typ wählen.



Im Verzeichniss "GlassFish" den GlassFish-Server auswählen. Je nach Eclipse-Version kann es sein, dass hier zwei Glassfish-Versionen zur Verfügung stehen (GlassFish 3.1, GlassFish 4). In diesem Fall muss "GlassFish 3.1" gewählt werden.

Übrige Angaben übernehmen und mit "Next" bestätigen.

4. Runtime Properties



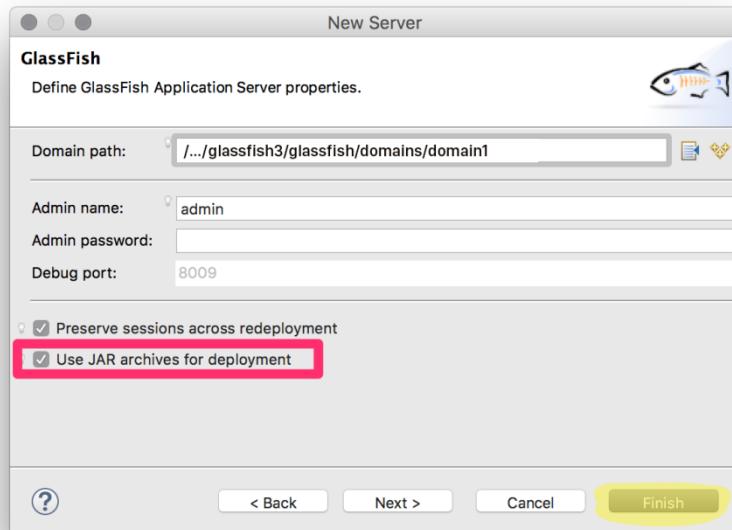
Unter "GlassFish location:" den Speicherort des Glassfish-Servers auf dem Computer angeben. Über den Browse-Button rechts kann je nach Speicherort in das Verzeichnis

.../glassfish3/glassfish navigiert werden.

Der Name wechselt automatisch auf GlassFish 3.1.

Übrige Angaben übernehmen und mit "Next" bestätigen.

5. Application Server Properties

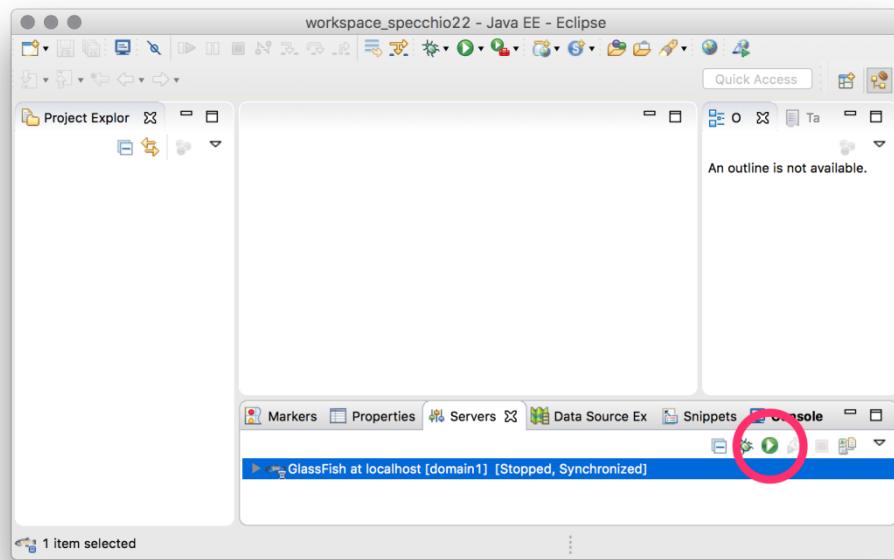


Der "Domain path" wird automatisch richtig übernommen.

Häkchen bei "Use JAR archives for deployment" setzen.

Mit "Finish" abschliessen.

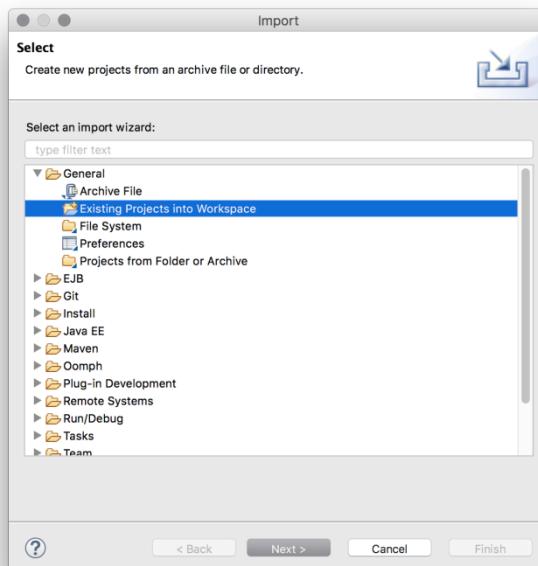
6. Server starten



Nach dem Startprozess befindet sich der Server im Status “Started”.

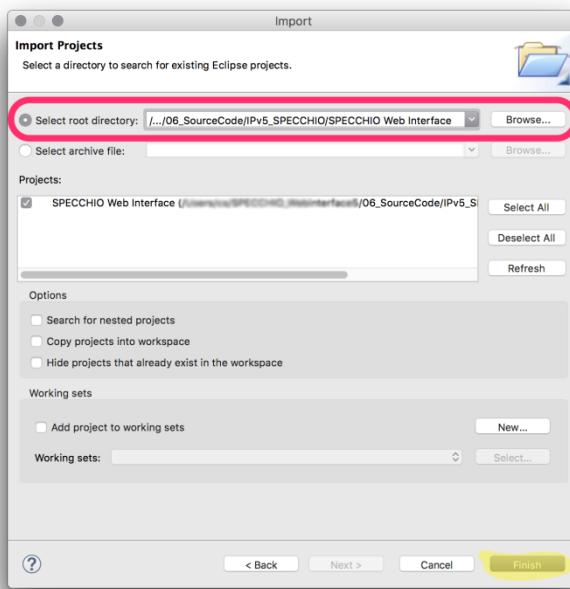
9.5.3 Projekt importieren

1. Neuer Import



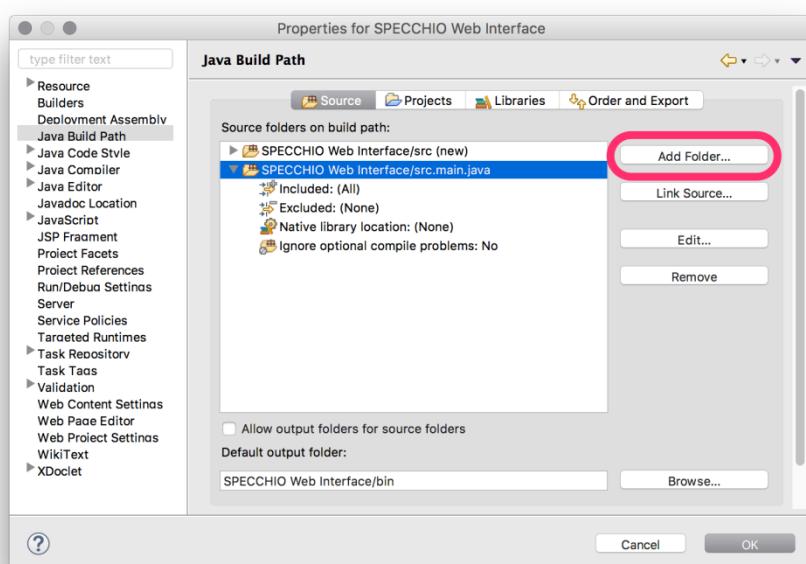
In Eclipse über das Menü File → Import... den Import-Dialog öffnen.
Verzeichnis “General” öffnen und “Existing Projects into Workspace” wählen.
Mit “Next” zum nächsten Schritt wechseln.

2. Dateipfad des Projekts wählen



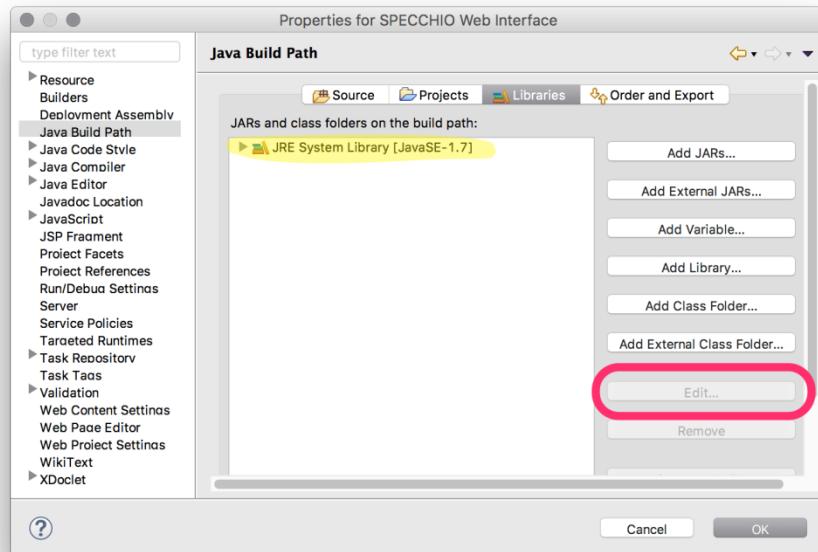
Das Projekt ist im Git-Repository unter
“.../06_SourceCode/IPv5_SPECCHIO/SPECCHIO Web Interface” abgelegt.
Mit “Finish” den Import-Prozess abschliessen.

3. Java Build Path konfigurieren: Source (Rechtsklick auf Projekt → Properties → Java Build Path → Source)



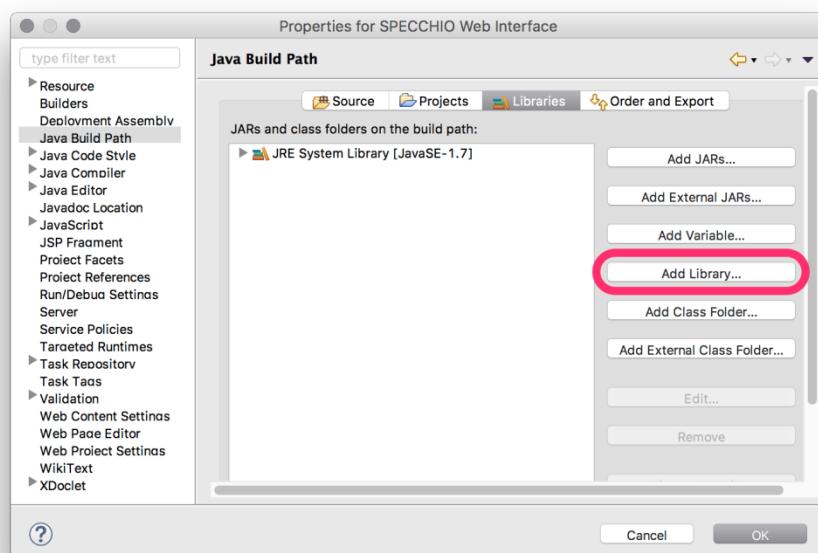
In der Rubrik “Source” über den Button “Add Folder” das Verzeichnis “SPECCHIO Web Interface/src.main.java” hinzufügen.

4. Java Build Path konfigurieren: JRE System Library 1.7
(Rechtsklick auf Projekt → Properties → Java Build Path → Libraries)

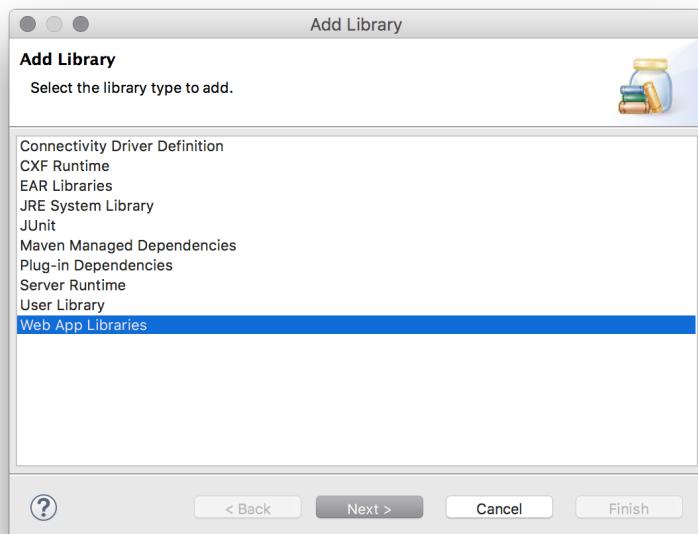


In der Rubrik “Libraries” über den Button “Edit” die korrekte JRE System Library wählen.
Es muss Java-Version 1.7 sein.

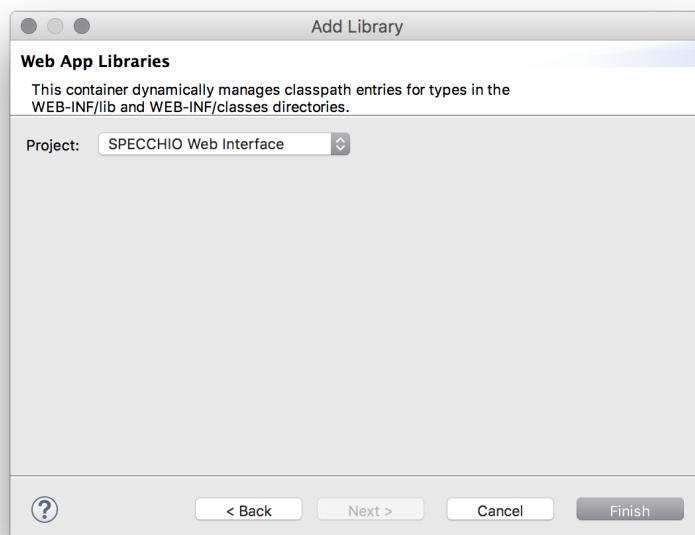
5. Java Build Path konfigurieren: Web App Libraries
(Rechtsklick auf Projekt → Properties → Java Build Path → Libraries)



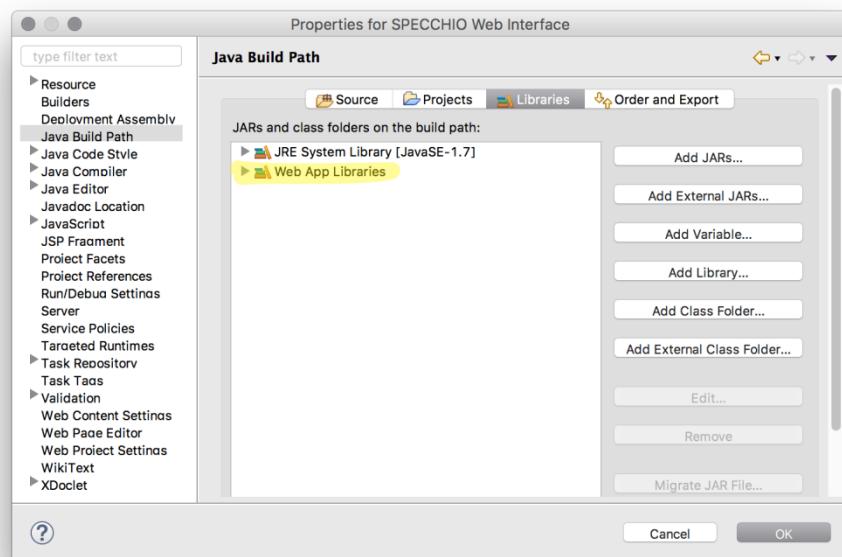
In der Rubrik “Libraries” den Button “Add Library...” wählen.



“Web App Libraries” wählen und mit “Next” bestätigen.

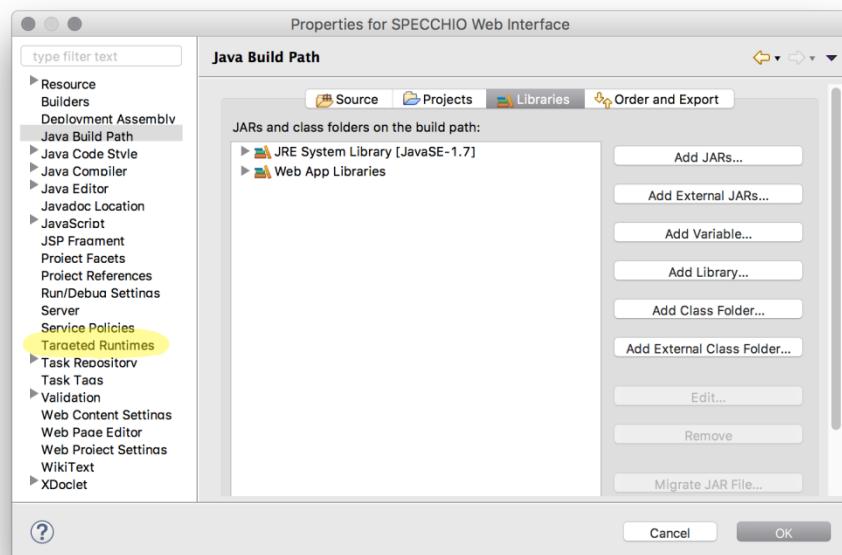


“SPECCHIO Web Interface” wählen und mit “Finish” abschliessen.

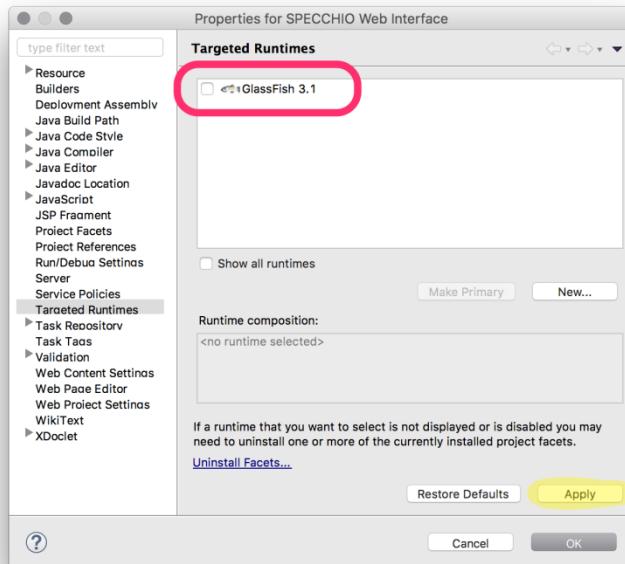


Die "Web App Libraries" wurden hinzugefügt. Mit "OK" bestätigen.

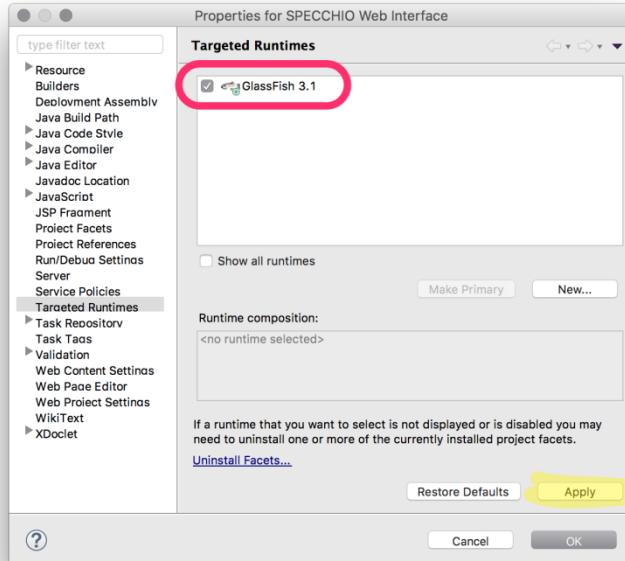
6. Java Build Path konfigurieren: GlassFish System Libraries
(Rechtsklick auf Projekt → Properties → Java Build Path → Libraries)



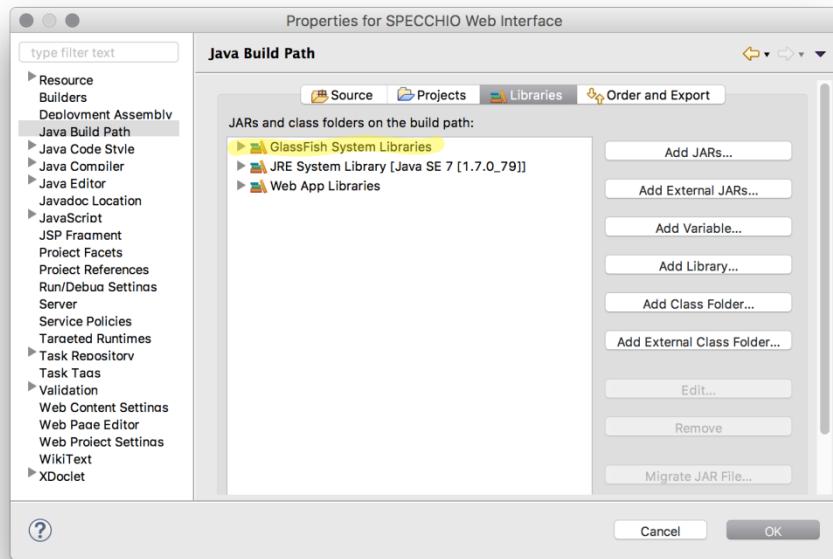
Um die "GlassFish System Libraries" hinzuzufügen muess zuerst zu den "Targeted Runtimes" gewechselt werden.



Hier sollte der Glassfish-Server aufgelistet sein. Wenn die Checkbox mit einem Häkchen ausgewählt ist, muss die Checkbox deaktiviert werden. Danach mit dem Button "Apply" bestätigen.



Das Häkchen beim Glassfish-Server erneut setzen und mit "Apply" bestätigen.



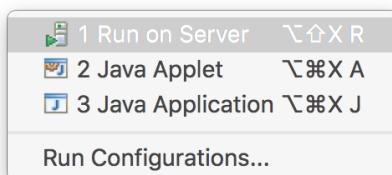
Unter "Java Build Path" wird nun die "GlassFish System Libraries" aufgeführt.

- Nachdem der Server erstellt und der Java Build Path konfiguriert wurden, sollten keine Errors mehr angezeigt werden. Das Projekt ist bereit, um gestartet zu werden.

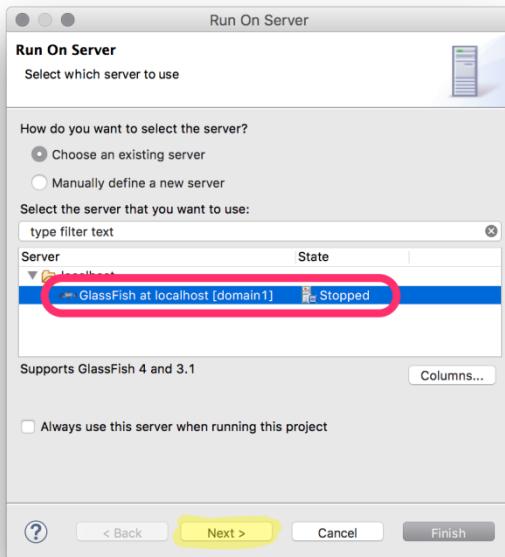
9.5.4 Projekt starten

Für den Start des Projekts werden folgende Bedingungen vorausgesetzt:

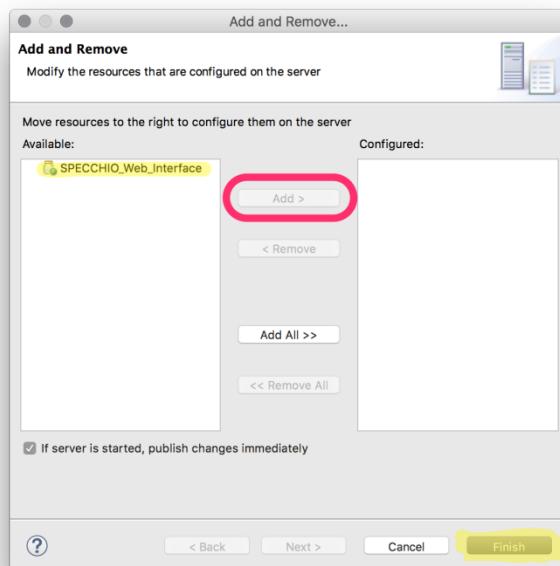
- Glassfish-Server in Eclipse erstellt
- Eclipse-Projekt importiert (keine Errors)
- SPECCHIO Virtualbox ist für den externen Zugang konfiguriert und läuft.



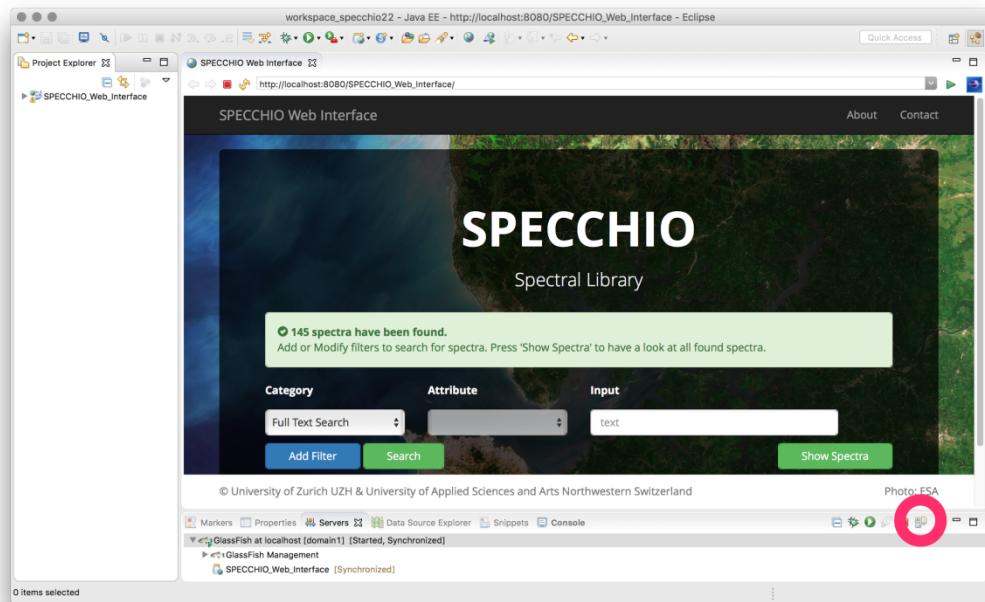
Mit Rechtsklick auf das Projekt in Eclipse kann das Projekt über "Run As -> 1 Run on Server" gestartet werden.



Es muss "GlassFish at localhost [domain1]" ausgewählt sein. Mit "Next" gelangt man zum nächsten Schritt.



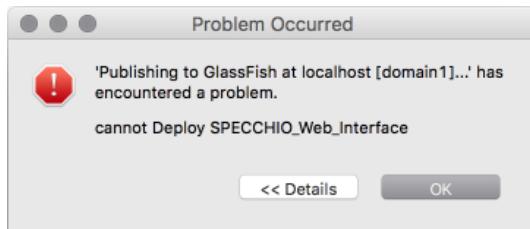
Falls nicht bereits geschehen, muss die Applikation zur Konfiguration hinzugefügt werden. Mit "Finish" abschliessen.



Die Applikation wird in Eclipse in einem internen Browser-Fenster angezeigt. Es können nun im Code in Eclipse Änderungen vorgenommen werden und über das “Publish”-Icon im Server-Panel auf dem Glassfish-Server aktualisiert werden.

Die Web-Applikation kann über die URL http://localhost:8080/SPECCHIO_Web_Interface/ in jedem beliebigen Browser betrachtet werden.

9.5.5 Probleme beim Starten des Projekts



Sollten Probleme beim Starten der Applikation aus Eclipse heraus entstehen, kann es daran liegen, dass auf dem Glassfish-Server bereits eine Applikation mit dem gleichen Namen vorliegt. In diesem Fall muss die vorhergehende Applikation sauber “undeployed” werden.

1. In einem beliebigen Browser die Glassfish-Administrations-Konsole über <http://localhost:4848> öffnen (Glassfish-Server muss in Eclipse gestartet sein).
2. Das Verzeichnis “Applications” öffnen.

3. Die entsprechende Applikation (hier: SPECCHIO_Web_Interface) "undeployen".

The screenshot shows the GlassFish Admin Console interface. On the left, there's a tree view with 'Applications' selected. In the main panel, under 'Deployed Applications (4)', the 'SPECCHIO_Web_Interface' row is selected. The 'Action' column for this row contains three buttons: 'Launch', 'Redeploy', and 'Reload'. Above the table, there are buttons for 'Deploy...', 'Undeploy', 'Enable', and 'Disable'. The 'Undeploy' button is highlighted with a red box.

Name	Enabled	Engines	Action
SPECCHIO_Web_Interface	✓	web	Launch Redeploy Reload
SPECCHIO_Web_Interface_1.1	✓	web	Launch Redeploy Reload
SampleServlet	✓	web	Launch Redeploy Reload
specchio_web_interface_neon2Test_01	✓	web	Launch Redeploy Reload

4. Nach dem "Undeploy"-Prozess kann die Applikation erneut aus Eclipse heraus gestartet werden. Der Fehler sollte behoben sein.

Ein weiteres mögliches Problem kann darin bestehen, dass im Projektnamen Leerzeichen enthalten sind. In diesem Fall müssen diese durch Underscore (_) ersetzt werden.
Rechtsklick auf Projekt → Refactor → Rename...

10 Deployment

Für den Betrieb der Produktionsumgebung ist der Auftraggeber verantwortlich. Ebenso für deren Sicherung/Backup. Um das SPECCHIO Web Interface von der Entwicklungsumgebung in die Produktionsumgebung zu deployen, sind nachfolgende Prozesse auszuführen.

10.1 WAR-Export

In Eclipse muss das Projekt als WAR-Datei exportiert werden. Mit Rechtsklick auf das Projekt → "Export" → "WAR file" kann die Web-Applikation als WAR-Datei exportiert werden.

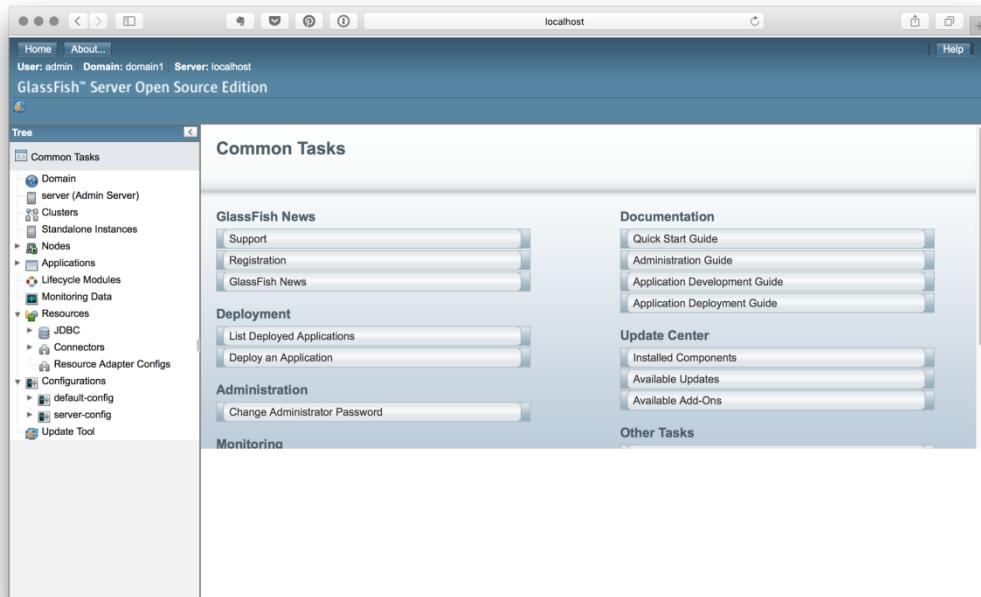


Die WAR-Datei kann an beliebigem Ort gespeichert werden. "Export source files" darf nicht ausgewählt sein. Mit "Finish" abschliessen.

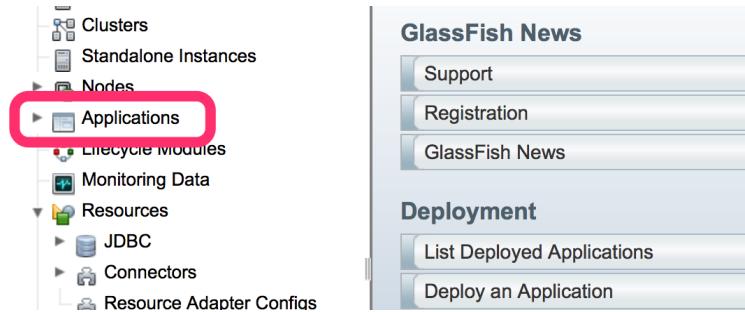
10.2 Deployment-Prozess

Die aus Eclipse exportierte WAR-Datei wird auf dem Webserver auf einem Glassfish-Server deployed.

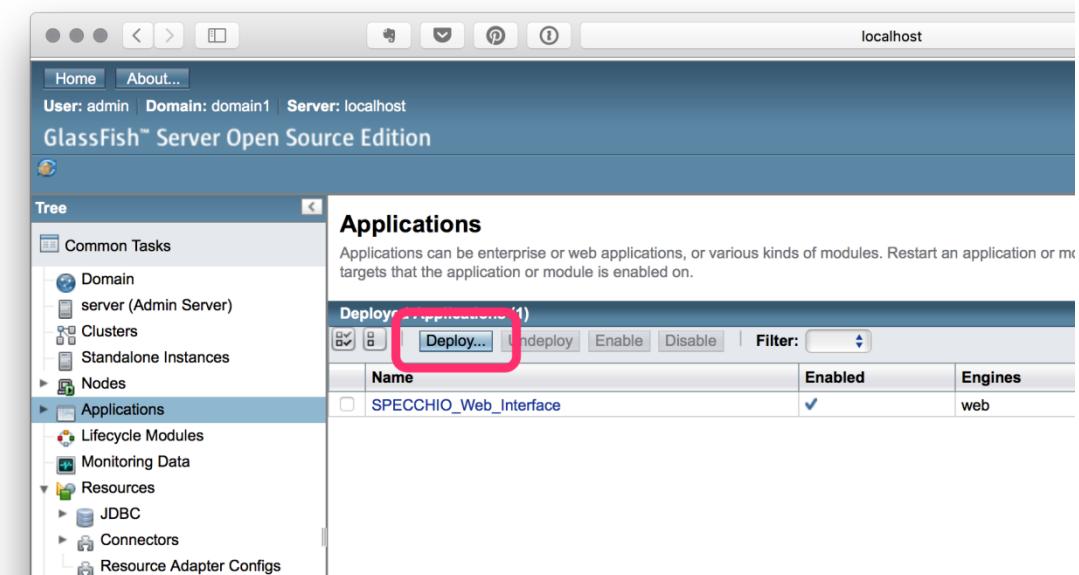
1. Glassfish-Administrations-Konsole via <http://localhost:4848> in einem beliebigen Browser öffnen (Glassfish-Server muss in Eclipse gestartet sein).



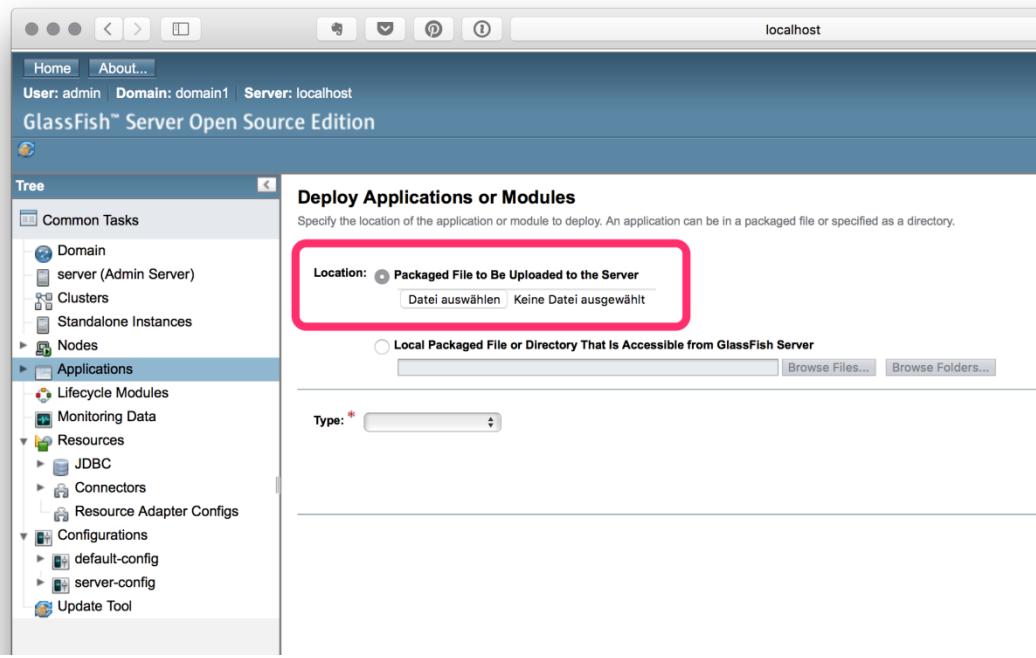
2. Das Verzeichnis "Applications" öffnen:



3. "Deploy..."-Button anklicken:

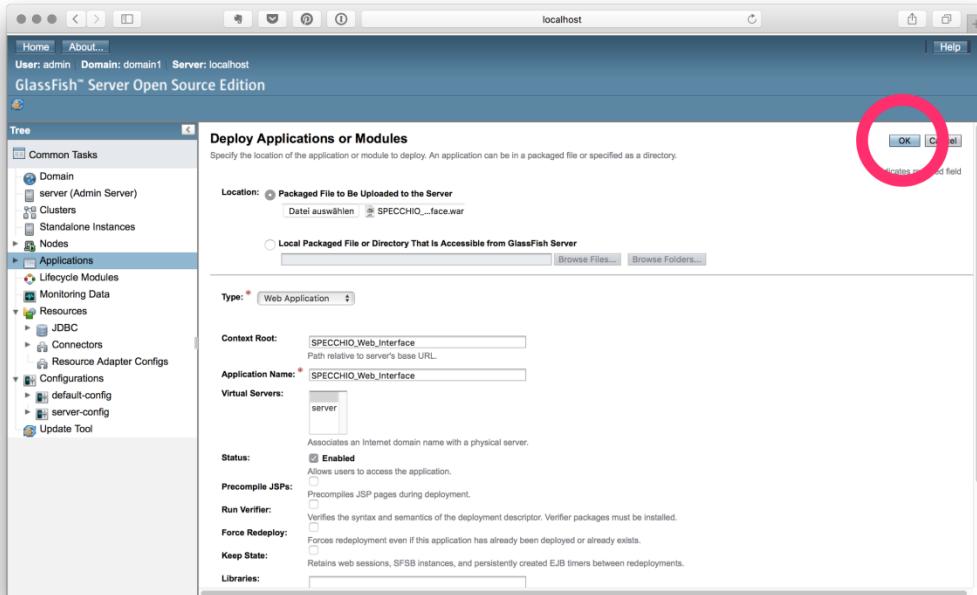


4. "Datei auswählen" anklicken:

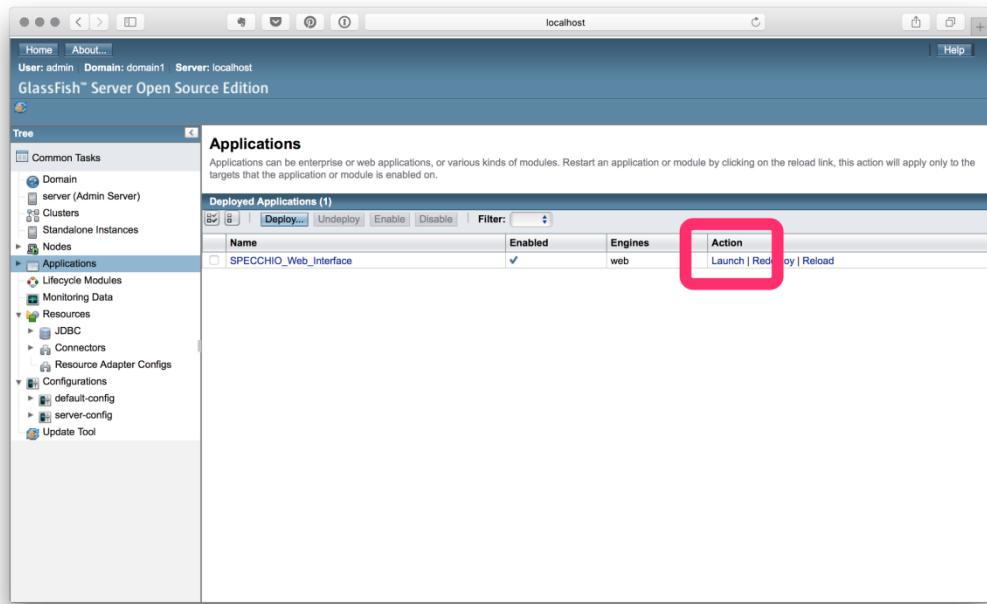


Wichtig: Der Dateiname des war-Files darf nicht identisch mit den bereits bestehenden war-Files sein. Ausserdem muss der Dateiname mit einem Buchstaben, Zahl oder Unterstrich beginnen und darf nur Buchstaben, Zahlen und folgende Zeichen enthalten: Bindestrich (-), Punkt (.), Unterstrich (_), Querstrich (/), Doppelpunkt (:), Rautenzeichen (#) und Semikolon (;).

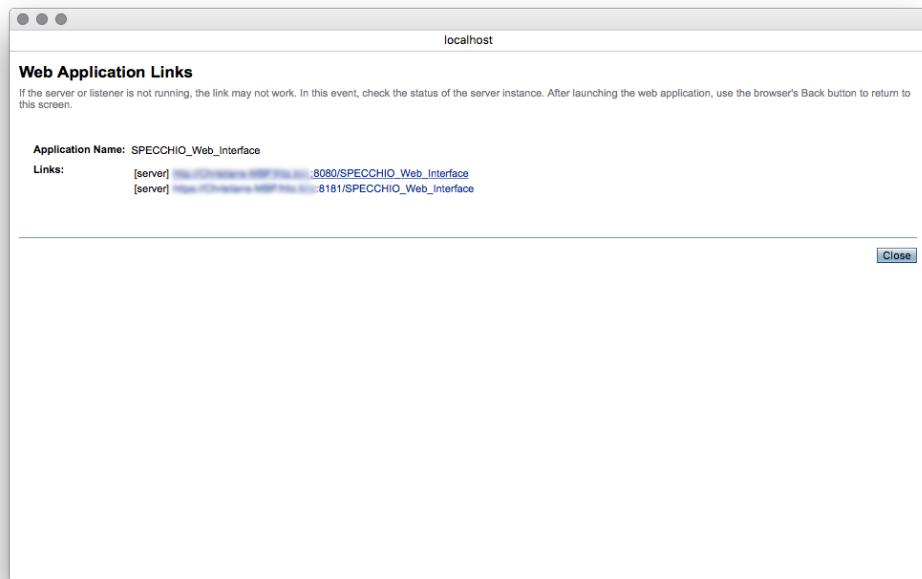
5. "OK" anklicken. Die Web-Applikation ist deployed:



6. "Launch" anklicken:



7. Der Applikations-Link wird angezeigt, welcher in einem beliebigen Browser eingesetzt werden kann.



10.3 Speicherort für SPECCHIO-Zugangsdaten (db_config.txt)

Die Zugangsdaten (db_config.txt) für die SPECCHIO-Datenbank müssen an einem bestimmten Ort innerhalb des Glassfish-Servers abgelegt werden. Der Speicherort wird aus Sicherheitsgründen hier nicht erwähnt. Der Pfad ist im Dokument mit den Zugangsdaten festgehalten. Dieses Dokument befindet sich auf dem Git-Repository unter .../08_Documentation/Zugangsdaten.docx.

11 Zugangsdaten

Alle Zugangsdaten befinden sich auf dem Git-Repository in einem passwortgeschützten Word-Dokument. Es ist unter .../08_Documentation/Zugangsdaten.docx abgelegt.

Die Zugangsdaten werden dem Auftraggeber übergeben. Aus Sicherheitsgründen wird das Passwort für die Word-Datei nur mündlich übergeben.

Dies Umfasst Zugangsdaten zu:

- Google Konto (Erstellt zur Anforderung eines Google Map API Keys)
- Pfad zum Ablageort des db_config.txt Files

12 Referenzen

- ESA:
http://www.esa.int/spaceinimages/Sets/Earth_observation_image_of_the_week
- Google Maps API:
https://enterprise.google.com/intl/de/maps/pricing/?utm_source=google&utm_medium=cpc&utm_campaign=2016-geo-emea-endor-gmedia-search-crossreg-&utm_content=ch%7Cde%7Chybr%7C1001878%7C%7Cbk%7Cbrand%7C%7C&gclid=Cj0KEQiAhZPDBRCz642XqYOCpb8BEiQANUcwTzk7SO0Z5dx4r_Vp3uKZ207cDm9o0lyuKCF216iHdkYaAuw68P8HAQ&dclid=CPSb4ffFmdECFRUYGwods3QFbA
- Highcharts.com: <http://www.highcharts.com>
- Oracle Java EE: <http://docs.oracle.com/javaee/6/tutorial/doc/geysj.html>
- Wikipedia: https://de.wikipedia.org/wiki/Web_Application_Archive