

# ISW: Software Engineering

## WS 2015/16

### GWT – Interaktionen

**Marcus Seiler**

Institute of Computer Science  
Chair of Software Engineering  
Im Neuenheimer Feld 326  
69120 Heidelberg, Germany

<http://se.ifi.uni-heidelberg.de>

[marcus.seiler@informatik.uni-heidelberg.de](mailto:marcus.seiler@informatik.uni-heidelberg.de)



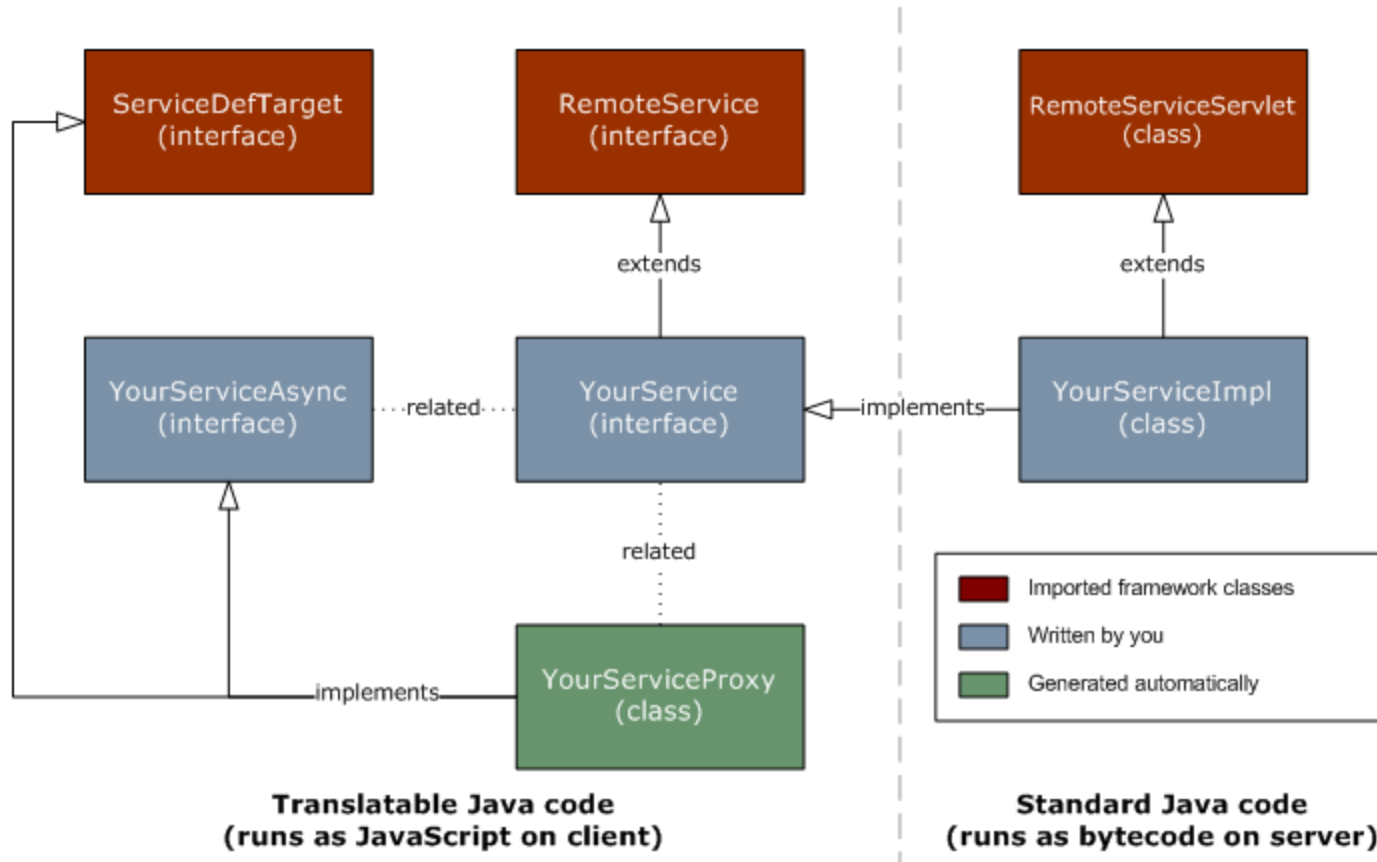
RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG

- Ausgangspunkt
  - Wir haben Oberflächenelemente, z.B. Buttons, Textfelder oder Tabellen
  
- Wir wollen
  - Interaktionen hinzufügen, d.h. es soll etwas passieren, wenn wir auf einen Button drücken
  
- GWT bietet verschiedene Handler und Updater
  - Buttons: Verwendung von ClickHandler
  - Textfelder: Verwendung von ValueChangeHandler
  - Tabellenzellen: Verwendung von FieldUpdater

# Interaktionen zwischen Client und Server

---

- Interaktion := Datentransfer zwischen Client und Server
- GWT verwendet Remote Procedure Calls (RPC)
- Eine RPC-basierte Kommunikation besteht aus drei Teilen
  - Service Definition
  - Service Implementierung
  - Service Erzeugung und Nutzung



- Interface zur Auflistung aller RPC-Operationen

```
@RemoteServiceRelativePath("bookservice")
public interface BookService extends RemoteService {

    public ArrayList<Book> loadBooks();

    public void saveBooks(ArrayList<Book> books);

}
```

- Asynchrones Interface für den Service

```
public interface BookServiceAsync {

    void loadBooks(AsyncCallback<ArrayList<Book>> callback);

    void saveBooks(ArrayList<Book> books, AsyncCallback<Void> callback);

}
```

## ■ Implementierung auf Serverseite

```
public class BookServiceImpl extends RemoteServiceServlet implements BookService {

    private static final long serialVersionUID = 2610052049222392123L;

    @Override
    public ArrayList<Book> loadBooks() {
        ArrayList<Book> books = new ArrayList<Book>();
        ...
        return books;
    }

    @Override
    public void saveBooks(ArrayList<Book> books) {
        ...
    }
}
```

- Um die Service Implementierung zu nutzen, muss in der Datei war/WEB-INF/web.xml die Service Implementierung eingetragen werden

```
...  
<servlet>  
<servlet-name>BookServiceImpl</servlet-name>  
<servlet-class>de.uhd.bookexample.server.BookServiceImpl</servlet-class>  
</servlet>  
<servlet-mapping>  
<servlet-name>BookServiceImpl</servlet-name>  
<url-pattern>/bookexample/bookservice</url-pattern>  
</servlet-mapping>  
...
```

- Ein Service kann im Client erzeugt werden

```
BookServiceAsync bookService = GWT.create(BookService.class);
```

- Anschließend können die RPC-Operationen aufgerufen werden

```
bookService.loadBooks(new AsyncCallback<ArrayList<Book>>() {  
  
    @Override  
    public void onSuccess(ArrayList<Book> result) {  
        // Do something with the result  
    }  
  
    @Override  
    public void onFailure(Throwable caught) {  
        GWT.log(caught.getMessage());  
    }  
});
```



- Live-Demo zum Mitmachen
  - Interaktion mit verschiedenen Handlern und Services
  - Webanwendung für Bücher
    - 04-gwt-book-example.zip aus Moodle herunterladen

- Offizielle GWT Dokumentation von Google
  - <http://www.gwtproject.org/doc/latest/tutorial/RPC.html>
  - <http://www.gwtproject.org/doc/latest/DevGuideServerCommunication.html>

---

## Marcus Seiler

Institute of Computer Science  
Chair of Software Engineering  
Im Neuenheimer Feld 326  
69120 Heidelberg, Germany

<http://se.ifi.uni-heidelberg.de>

[marcus.seiler@informatik.uni-heidelberg.de](mailto:marcus.seiler@informatik.uni-heidelberg.de)



RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG

---