

Invoking the REST Service



Antonio Goncalves

JAVA CHAMPION

@agoncal www.antoniogoncalves.org



Previous Module



Navigate between components

Angular Router

Simple navigation

Passing parameters to the router

Executing an action before routing

Mock components with fake data

Overview



Angular Services

Model

Observables

Code generated by Swagger CodeGen

Invoke Java EE back-end

BookStore runs from front to end



What Is a Service?

TypeScript class processing
business logic

No graphical representation

Share data and behavior
across multiple components

Injected into components



A Service

```
import { Injectable } from '@angular/core';
```

```
@Injectable()
```

```
export class BookService {
```

```
    public countBooks(): number {
```

```
        return 3;
```

```
    }
```

```
}
```



A Component Injecting a Service

```
@Component({  
  selector: 'bs-book-list',  
  templateUrl: './book-list.component.html'  
})  
export class BookListComponent implements OnInit {  
  
  private nbBooks: number;  
  
  constructor(private bookService: BookService) { }  
  
  ngOnInit() {  
    this.nbBooks = this.bookService.countBooks();  
  }  
}
```



A Service

```
import { Injectable } from '@angular/core';
```

```
@Injectable()
```

```
export class BookService {
```

```
  public countBooks(): number {
```

```
    return 3;
```

```
  }
```

```
}
```



A Service with HTTP

```
import { Injectable } from '@angular/core';  
import { Http }       from '@angular/http';
```

```
@Injectable()  
export class BookService {  
  
  constructor(private http: Http) { }
```

```
  public countBooks(): number {  
    return this.http.get('/books/count');  
  }  
}
```



A Service with Observable

```
import { Injectable } from '@angular/core';  
import { Http }       from '@angular/http';  
import 'rxjs/Rx';
```

```
@Injectable()  
export class BookService {  
  
    constructor(private http: Http) { }  
  
    public countBooks(): Observable<number> {  
        return this.http.get('/books/count');  
    }  
}
```



What Is an Observable?



Reactive programming

RxJS, RxJava...

Event-driven

Subscribe

Combine, filter, group... events

Angular built on reactive programming

Observable

```
Observable.from([1, 2, 3, 4, 5])
```

```
.subscribe(x => console.log(x));
```



Map

```
Observable.from([1, 2, 3, 4, 5])  
  .map(x => 10 * x)  
  
  .subscribe(x => console.log(x));
```



Filter

```
Observable.from([1, 2, 3, 4, 5])  
  .map(x => 10 * x)  
  .filter(x => x > 25)  
  .subscribe(x => console.log(x));
```



The BookStore Service and Model



Book service

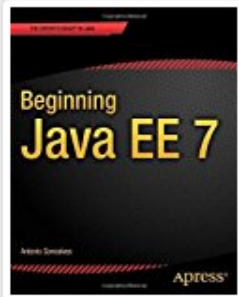
Book model

Lists all the books

Displays the details of a book

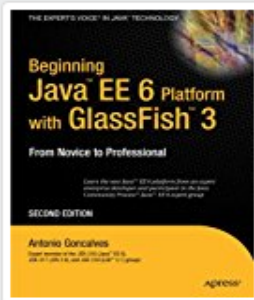
Form to create a new book

List all the books + 3



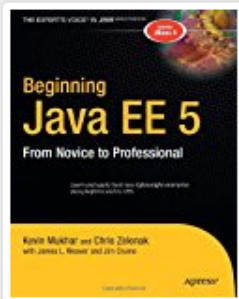
Beginning Java EE 7

Java enterprise edition (java ee) continues to be one of the leading java technologies and platforms. Beginning java ee 7 is the first tutorial book on java ee 7. Step by step and easy to follow, this book describes many of the java ee 7 specifications and reference implementations and shows them in action using practical examples. This definitive book also uses the newest version of glassfish to deploy and administer the code examples. Java ee 7 environment context and dependency injection bean validation java persistence api object-relational mapping managing persistent o...



Beginning Java EE 6 with GlassFish 3

Java Enterprise Edition (Java EE) continues to be one of the leading Java technologies and platforms from Oracle (previously Sun). Beginning Java EE 6 Platform with GlassFish 3, Second Edition is this first tutorial book on the final version of the Java EE 6 Platform. Step by step and easy to follow, this book describes many of the Java EE 6 specifications and reference implementations, and shows them in action using practical examples. This book uses the new version of GlassFish 3 to deploy and administer the code examples. Written by an expert member of the Java EE 6 spec...



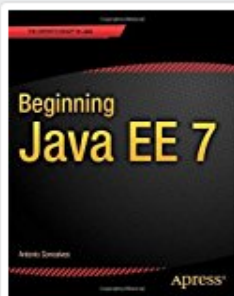
Beginning Java EE 5: From Novice to Professional

Sun's new lightweight Java Enterprise Edition (EE) 5 is an extremely powerful platform for developing enterprise-level Java-based applications, primarily for the server. This book shows you how to harness that power, examining how the pieces of the new Java EE 5 platform fit together, including the redesigned annotations-driven EJB 3 spec as well as JavaServer Faces (JSF), integrated into the platform for the first time. Hands-on tutorials are also included, along with clear explanations and working code examples. You will grow to take the next step—from writing client-side...

GET

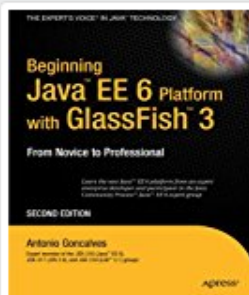
<http://www.cdbookstore.com/books/count>

List all the books - 3



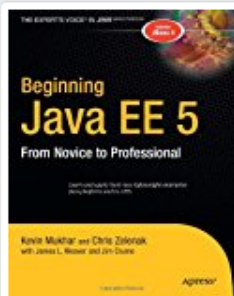
Beginning Java EE 7

Java enterprise edition (java ee) continues to be one of the leading java technologies and platforms. Beginning java ee 7 is the first tutorial book on java ee 7. Step by step and easy to follow, this book describes many of the java ee 7 specifications and reference implementations and shows them in action using practical examples. This definitive book also uses the newest version of glassfish to deploy and administer the code examples. Java ee 7 environment context and dependency injection bean validation java persistence api object-relational mapping managing persistent o...



Beginning Java EE 6 with GlassFish 3

Java Enterprise Edition (Java EE) continues to be one of the leading Java technologies and platforms from Oracle (previously Sun). Beginning Java EE 6 Platform with GlassFish 3, Second Edition is this first tutorial book on the final version of the Java EE 6 Platform. Step by step and easy to follow, this book describes many of the Java EE 6 specifications and reference implementations, and shows them in action using practical examples. This book uses the new version of GlassFish 3 to deploy and administer the code examples. Written by an expert member of the Java EE 6 spec...



Beginning Java EE 5: From Novice to Professional

Sun's new lightweight Java Enterprise Edition (EE) 5 is an extremely powerful platform for developing enterprise-level Java-based applications, primarily for the server. This book shows you how to harness that power, examining how the pieces of the new Java EE 5 platform fit together, including the redesigned annotations-driven EJB 3 spec as well as JavaServer Faces (JSF), integrated into the platform for the first time. Hands-on tutorials are also included, along with clear explanations and working code examples. You will grow to take the next step—from writing client-side...

GET

<http://www.cdbookstore.com/books>

Detail of a book

| | | |
|-----------------|---|----|
| Title | Beginning Java EE 7 | |
| Description | Java enterprise edition (java ee) continues to be one of the leading java technologies and platforms. Beginning java ee 7 is the first tutorial book on java ee 7. Step by step and easy to follow, this book describes many of the java ee 7 specifications and reference implementations and shows them in action using practical examples. This definitive book also uses the newest version of glassfish to deploy and administer the code examples. Java ee 7 environment context and dependency injection bean validation java persistence api object-relational mapping managing persistent object enterprise java beans callbacks, timer service and authorization interceptors and transactions javaserver faces processing and navigation xml and json messaging soap web services restful web service table of contents: introduction to the ejb 3 architecture session beans entities and the java persistence api advanced persistence features message-driven beans ejb and web services integrating session beans, entities, message-driven beans and web services transaction management ejb performance and testing context and dependency injection (cdi) packaging and deployment ejb client applications ejb runtime environments | |
| Unit cost | 9,99 | \$ |
| ISBN | 8132214080 | |
| Number of pages | 29 | |
| Language | ENGLISH | |

Delete

Back

GET

http://www.cdbookstore.com/books/1234

Detail of a book

| | | |
|-----------------|---|----|
| Title | Beginning Java EE 7 | |
| Description | Java enterprise edition (java ee) continues to be one of the leading java technologies and platforms. Beginning java ee 7 is the first tutorial book on java ee 7. Step by step and easy to follow, this book describes many of the java ee 7 specifications and reference implementations and shows them in action using practical examples. This definitive book also uses the newest version of glassfish to deploy and administer the code examples. Java ee 7 environment context and dependency injection bean validation java persistence api object-relational mapping managing persistent object enterprise java beans callbacks, timer service and authorization interceptors and transactions javaserver faces processing and navigation xml and json messaging soap web services restful web service table of contents: introduction to the ejb 3 architecture session beans entities and the java persistence api advanced persistence features message-driven beans ejb and web services integrating session beans, entities, message-driven beans and web services transaction management ejb performance and testing context and dependency injection (cdi) packaging and deployment ejb client applications ejb runtime environments | |
| Unit cost | 9,99 | \$ |
| ISBN | 8132214080 | |
| Number of pages | 29 | |
| Language | ENGLISH | |

Delete

Back

DELETE <http://www.cdbookstore.com/books/1234>

Create a new book

Title

Beginning Java EE 7

Description

Java enterprise edition (Java EE) continues to be one of the leading java technologies and platforms. Beginning Java EE 7 is the first tutorial book on Java EE 7. Step by step and easy to follow, this book describes many of the Java EE 7 specifications and reference implementations and shows them in action using practical examples. This definitive book also uses the newest version of Glassfish to deploy and administer the code examples. Java EE 7 environment context and dependency injection bean validation java persistence api object-relational mapping managing persistent object enterprise java beans callbacks, timer service and authorization interceptors and transactions JavaServer Faces processing and navigation xml and JSon messaging soap web services restful web service table of contents: introduction to the EJB 3 architecture session beans entities and the java persistence api advanced persistence features message-driven beans EJB and web services integrating session beans, entities, message-driven beans and web services transaction management EJB performance and testing context and dependency injection (cdi) packaging and deployment EJB client applications EJB runtime environments

Unit cost

9.99

\$

Number of pages

345

Image URL

Language

ENGLISH

Create

Cancel

POST http://www.cdbookstore.com/books

```
@Injectable()
export class BookService {
  protected basePath =
    'http://localhost:8080/bookstore-back/api';

  public getBook(id: number):
    Observable<Book> {...}
}
```

```
{
  "swagger": "2.0",
  "info": {"description": "BookStore"},
  "host": "localhost:8080",
  "basePath": "/bookstore-back/api",
  "schemes": [ "http", "https" ],
  "paths": {
    "/books/{id}": {
      "get": {
        "operationId": "getBook",
        "produces": [ "application/json" ],
        "parameters": [ {"name": "id",
                          "type": "integer",
                          "pattern": "\\d+"} ],
        "summary": "Returns a book",
        "responses": {
          "404": { "description": "Not found" },
          "200": {
            "description": "Book found",
            "schema": {
              "$ref": "#/definitions/Book"
            }
          }
        }
      }
    }
  }
}
```



```
export class Book {  
  
  id?: number;  
  title: string;  
  description?: string;  
  unitCost?: number;  
  
  // ...  
}
```

```
"definitions" : {  
  "Book" : {  
    "description" : "Book representation",  
    "type" : "object",  
    "properties" : {  
      "id" : {  
        "type" : "integer",  
        "description" : "Identifier"  
      },  
      "title" : {  
        "type" : "string",  
        "description" : "Title of the book"  
      },  
      "description" : {  
        "type" : "string",  
        "description" : "Describes the book"  
      },  
      "unitCost" : {  
        "type" : "number",  
        "format" : "float",  
        "description" : "Unit cost"  
      }  
    }  
  }  
}
```



```

@Injectable()
export class BookService {
  protected basePath =
    'http://localhost:8080/bookstore-back/api';

  constructor(protected http: Http) { }

  public getBook(id: number):
    Observable<Book> {...}

  public countBooks(): Observable<number> { }

  public createBook(body: Book):
    Observable<{}> {...}

  public deleteBook(id: number):
    Observable<{}> {...}

  public getBooks():
    Observable<Array<Book>> {}
}

```

```

{
  "swagger": "2.0",
  "info": {"description": "BookStore"},
  "host": "localhost:8080",
  "basePath": "/bookstore-back/api",
  "schemes": [ "http", "https"],
  "paths": {
    "/books/{id}": {
      "get": {
        "operationId": "getBook",
        "produces": ["application/json"],
        "parameters": [{ "name": "id",
                          "type": "integer",
                          "pattern": "\\d+"}],
        "summary": "Returns a book",
        "responses": {
          "404": { "description": "Not found"},
          "200": {
            "description": "Book found",
            "schema": {
              "$ref": "#/definitions/Book"
            }
          }
        }
      }
    }
  }
}

```



Demo



Swagger CodeGen

Parse JSON contract

Generate service and model

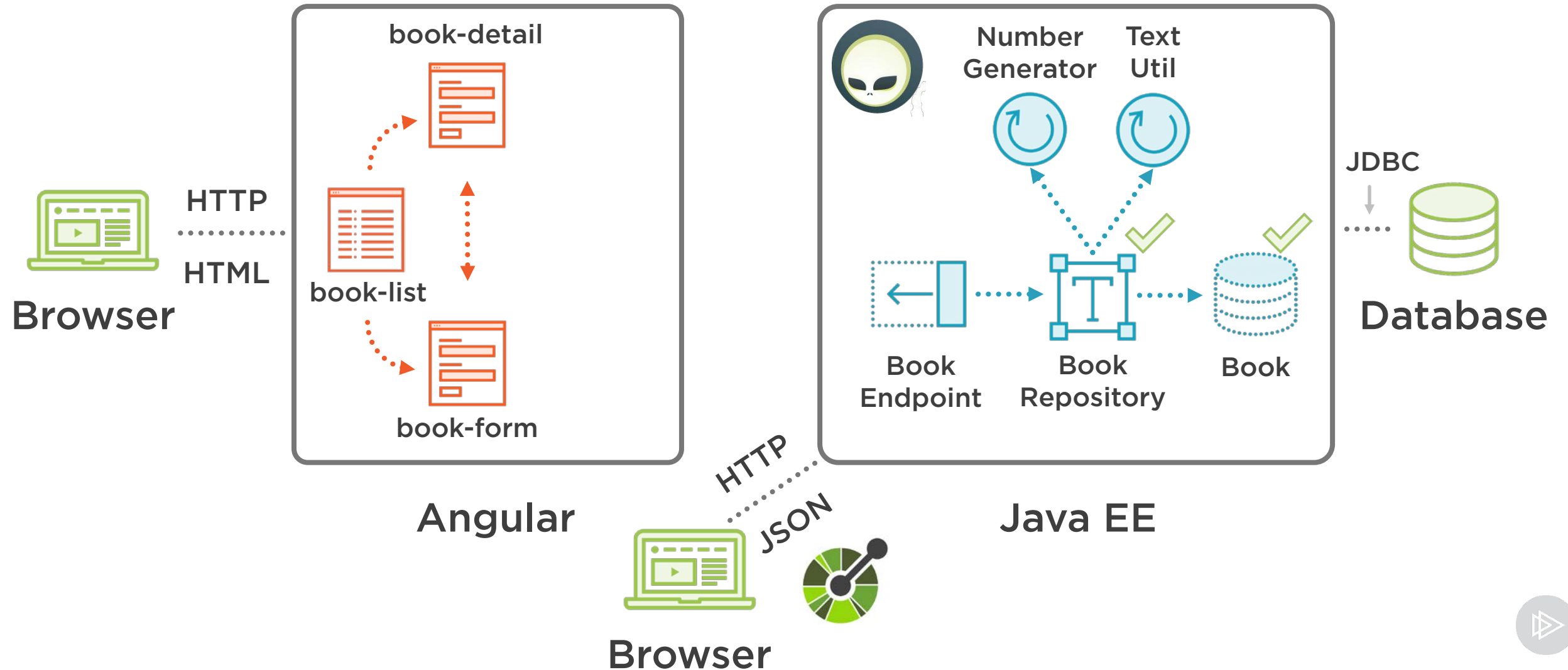
Invoke the Java EE Back-end

HTTP requests

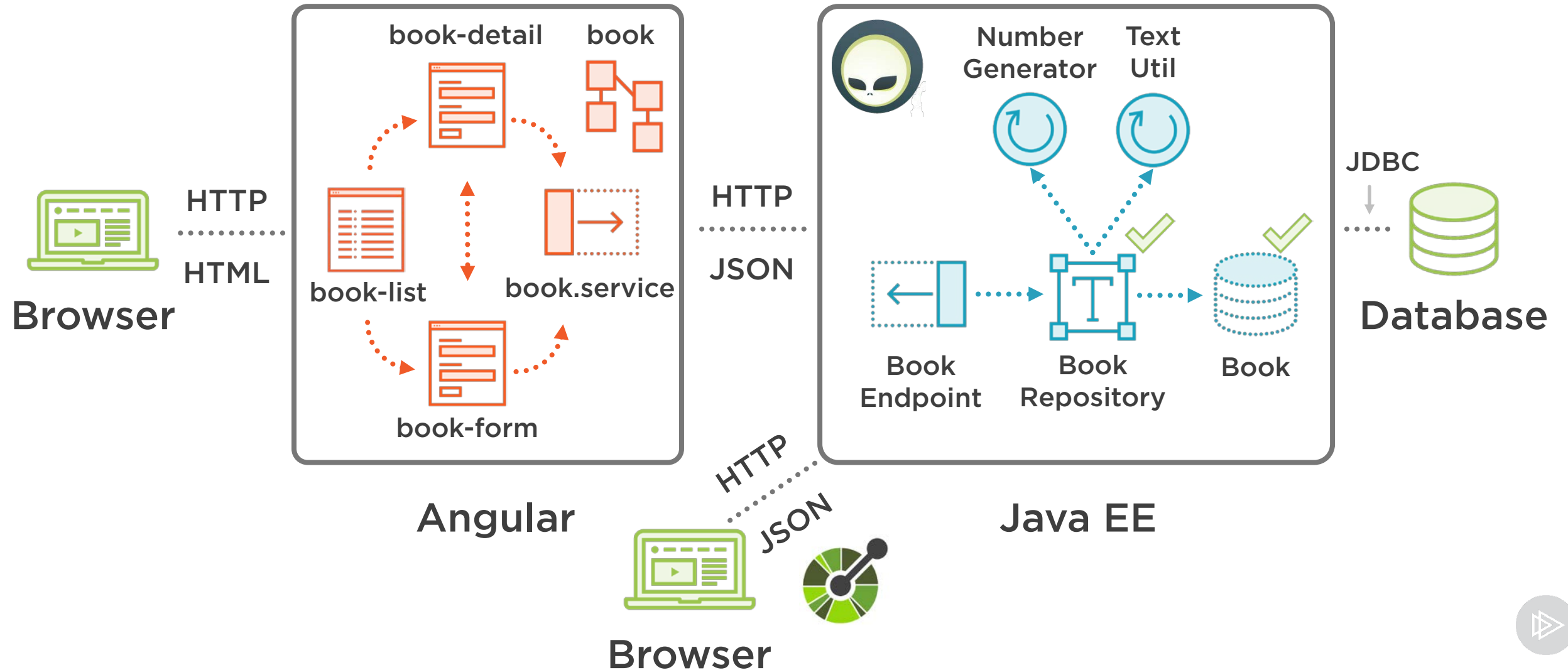
Observable



Anatomy of the BookStore Application



Anatomy of the BookStore Application



Summary



Understanding services

Understanding observables

Reactive programming

Swagger CodeGen

JSON contract

Generate service and model

HTTP to invoke back-end



Next Module



Revisit the application setup

Focus on application anatomy changes

Java EE back-end

Angular front-end

External references

