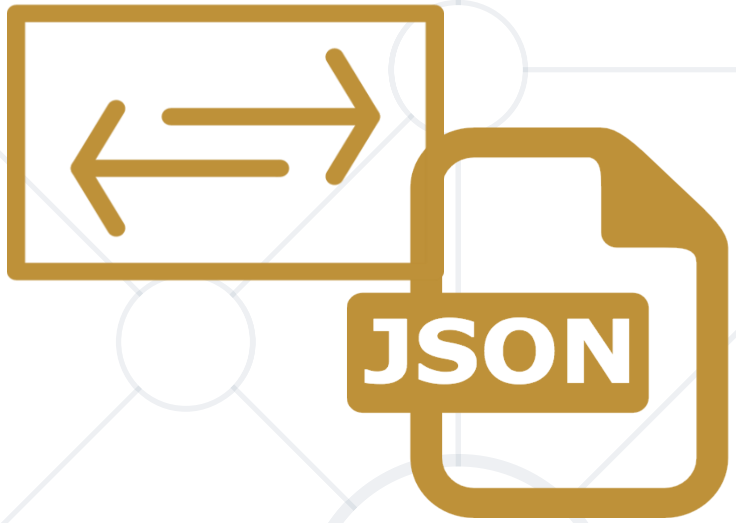
JSON

Exporting and Importing Data from JSON format



SoftUni Team

Technical Trainers

Software University

http://softuni.bg

Table of Content

Questions

SoftUni Foundation

sli.do

#JavaDb

JSON

Transmitting data objects via attribute-value pairs

JSON

- **JavaScript Object Notation**
  - Human-readable format to transmit **data objects** consisting of **attribute–value pairs** and **arrays**
  - Subset of JavaScript syntax
- Supports several data types:
  - Number, String, Boolean, Array, Object, null

JSON Example

person.json

```
{
  "firstName": "Daniel",
  "lastName": "Sempre",
  "age": 24,
  "isMarried": true
}
```

Key

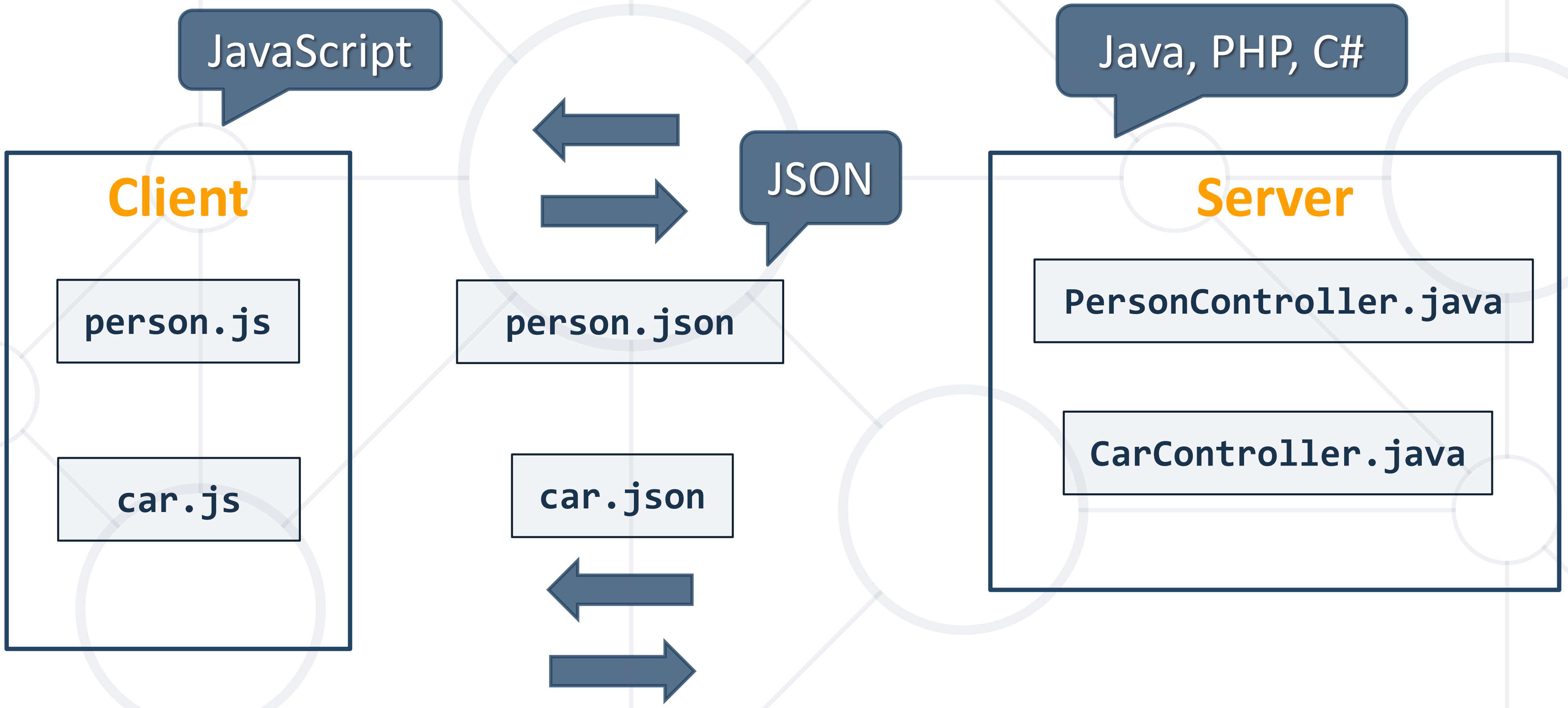
Value

student.json

```
{
  "firstName": "Daniel",
  "lastName": "Sempre",
  "age": 24,
  "courses": [
    {
      "name": "Java DB",
    },
    {
      "name": "HTML",
    },
  ]
}
```

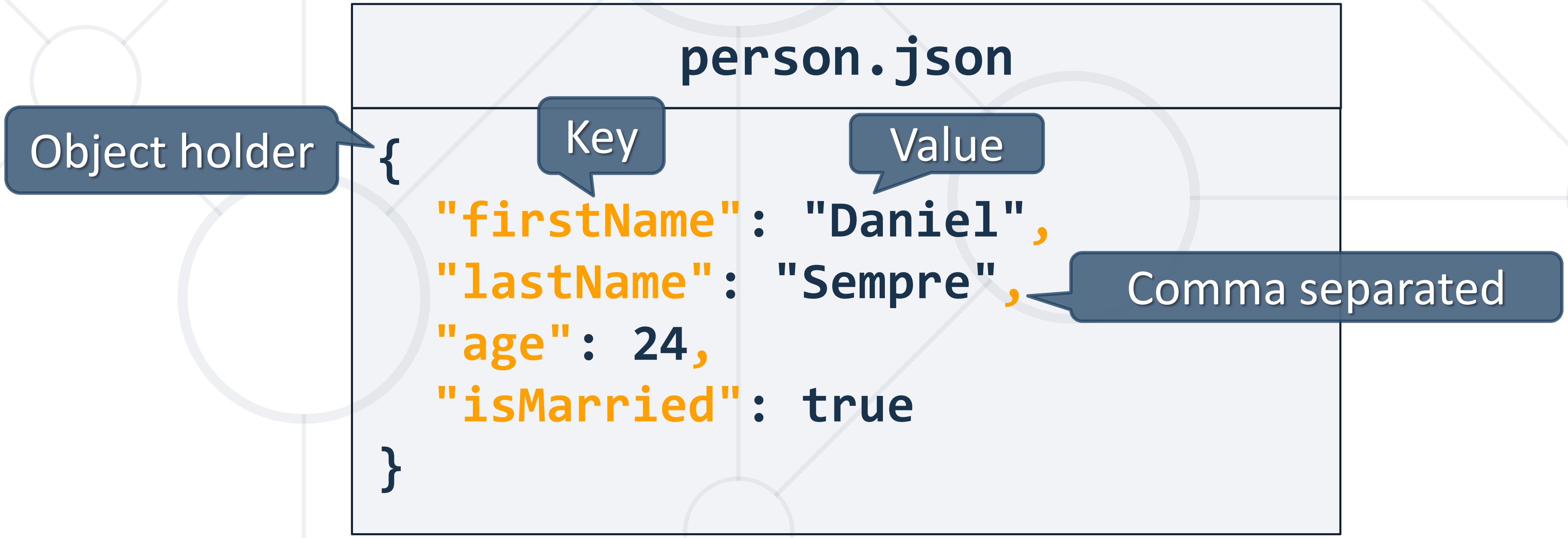
Array type value

JSON Function



JSON Structure

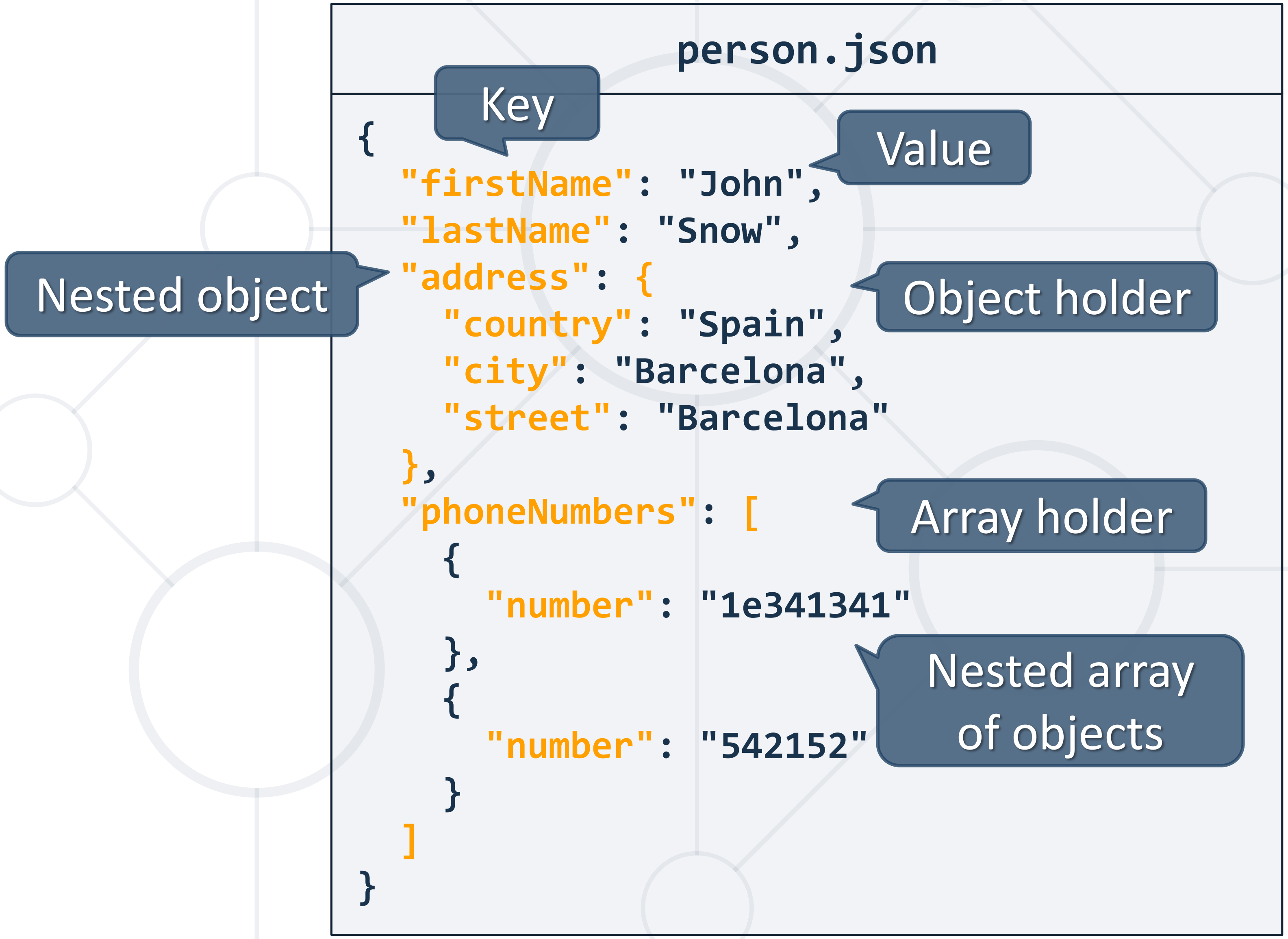
- Data is represented in **name/value** pairs
- Curly braces hold objects
- Square brackets hold **arrays**



person.json

```json
{
  "firstName": "Daniel",
  "lastName": "Sempre",
  "age": 24,
  "isMarried": true
}
```

JSON Structure



person.json

```
{
  "firstName": "John",
  "lastName": "Snow",
  "address": {
    "country": "Spain",
    "city": "Barcelona",
    "street": "Barcelona"
  },
  "phoneNumbers": [
    {
      "number": "1e341341"
    },
    {
      "number": "542152"
    }
  ]
}
```

GSON

Serialize and de-serialize objects with Java

GSON

- Provides easy to use mechanisms to convert **Java to JSON** and vice-versa
  - Originally developed by Google
- Generates compact and readability JSON output

pom.xml

```
<dependency>
            <groupId>com.google.code.gson</groupId>
            <artifactId>gson</artifactId>
</dependency>
```

GSON Initialization

- Gson objects are responsible for the JSON manipulations
  - GsonBuilder creates an instance of GSON
  - excludeFieldsWithoutExposeAnnotation() – excludes fields without **@Expose** annotation
  - setPrettyPrinting() – aligns and justifies the created JSON format
  - create() – creates an instance of Gson

JsonParser.java

```
Gson gson = new GsonBuilder()
                .excludeFieldsWithoutExposeAnnotation()
                .setPrettyPrinting()
                .create();
```

Export Single Object to JSON

AddressJsonDto.java

```
public class AddressJsonDto implements Serializable {

    @Expose
    private String country;

    @Expose
    private String city;

    @Expose
    private String street;
}
```

The field will be imported/exported

JsonParser.java

```
AddressJsonDto addressJsonDto = new AddressJsonDto();
addressJsonDto.setCountry("Bulgaria");
addressJsonDto.setCity("Sofia");
addressJsonDto.setStreet("Mladost 4");
String content = this.gson.toJson(addressJsonDto);
```

Creates JSON

Export Single Object to JSON

JsonParser.java

```java
AddressJsonDto addressJsonDto = new AddressJsonDto();
addressJsonDto.setCountry("Bulgaria");
addressJsonDto.setCity("Sofia");
addressJsonDto.setStreet("Mladost 4");
String content = this.gson.toJson(addressJsonDto);
```

address.json

```json
{
  "country": "Bulgaria",
  "city": "Sofia",
  "street": "Mladost 4"
}
```

Export Multiple Object to JSON

JsonParser.java

```java
List<AddressJsonDto> addressJsonDtos = new ArrayList<>();
addressJsonDtos.add(addressJsonDtoBulgaria);
addressJsonDtos.add(addressJsonDtoSpain);
String content = this.gson.toJson(addressJsonDtos);
```

addresses.json

```json
[
  {
    "country": "Bulgaria",
    "city": "Sofia",
    "street": "Mladost 4"
  },
  {
    "country": "Spain",
    "city": "Barcelona",
    "street": "Las Ramblas"
  }
]
```

Import Single Object to JSON

AddressJsonDto.java

```
public class AddressJsonDto implements Serializable {

    @Expose
    private String country;

    @Expose
    private String city;

    @Expose
    private String street;
}
```

The field will be imported/exported

JsonParser.java

```
AddressJsonDto addressJsonDto =
                this.gson.fromJson(AddressJsonDto.class, "/files/input/json/address.json");
```

Import Single Object to JSON

AddressJsonDto.java

```java
public class AddressJsonDto implements
Serializable {

    @Expose
    private String country;

    @Expose
    private String city;

    @Expose
    private String street;
}
```

address.json

```json
{
  "country": "Bulgaria",
  "city": "Sofia",
  "street": "Mladost 4"
}
```

Import Multiple Object to JSON

JsonParser.java

```java
AddressJsonDto[] addressJsonDtos =
                this.gson.fromJson(AddressJsonDto[].class, "/files/input/json/addresses.json");
```

Object Array

addresses.json

```json
[
  {
    "country": "Bulgaria",
    "city": "Sofia",
    "street": "Mladost 4"
  },
  {
    "country": "Spain",
    "city": "Barcelona",
    "street": "Las Ramblas"
  }
]
```

Summary

- JSON is a very easy to use and understand format
- GSON is a java library to operate with JSON files
  - Easy import and export

Questions?

SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SBTech
we know sports

telenor

SoftwareGroup
doing it right

NETPEAK

SmartIT

Postbank
Решения за твоето утре

SUPER HOSTING .BG

INDEAVR
Serving the high achievers

INFRAGISTICS

LIEBHERR

æternity

codexio

SoftUni Organizational Partners

SoftUni Foundation

ИНФОРМАЦИОННО ОБСЛУЖВАНЕ

OneBit SOFTWARE

WORLD OF MYTHS

Lukanet.com

codexio

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License