UNIVERSITÀ DELLA CALABRIA

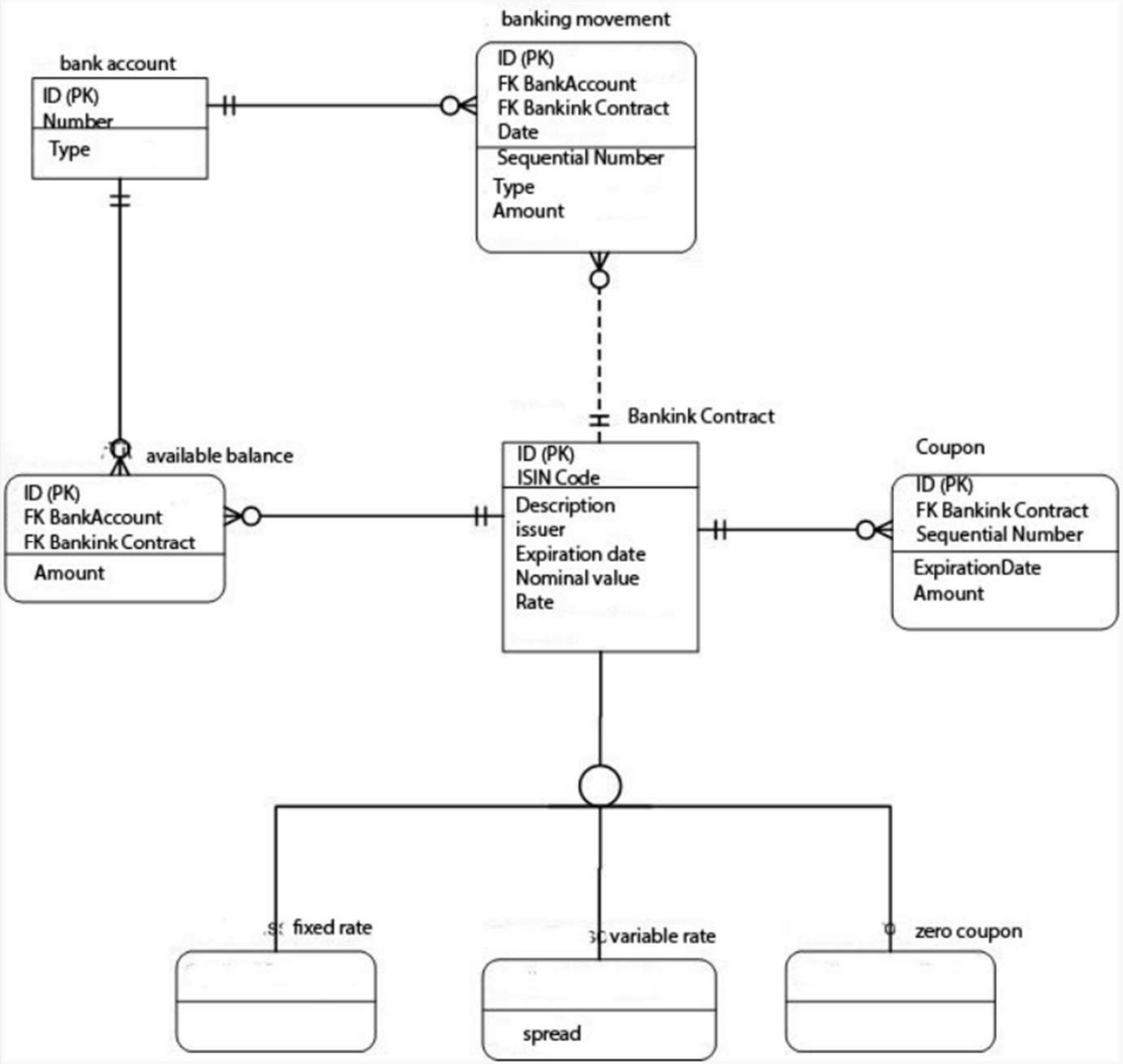
DIPARTIMENTO DI **MATEMATICA E INFORMATICA**

AGILE SOFTWARE DEVELOPMENT

Dr. Francesco Scalzo - francesco.scalzo@unical.it
Enterprise Solution Engineer at **Revelis s.r.l.**

Office hours & info: send a mail

Exercise


banking movement
bank account
ID (PK)
Number
Type
ID (PK)
FK BankAccount
FK Bankink Contract
Date
Sequential Number
Type
Amount
Bankink Contract
available balance
ID (PK)
FK BankAccount
FK Bankink Contract
Amount
ID (PK)
ISIN Code
Description
issuer
Expiration date
Nominal value
Rate
Coupon
ID (PK)
FK Bankink Contract
Sequential Number
ExpirationDate
Amount
fixed rate
variable rate
zero coupon
spread

Spring Project

PRESENTATION
- Angular

REST
- Spring - @Controller

BUSINESS
- Spring - @Service

DATA ACCESS
- Spring – JPA, @Repository

DATA
- DB – MariaDB / HSQLDB

- https://www.jetbrains.com/idea/download/
- https://start.spring.io/

- REST has quickly become the de-facto standard for building web services on the web because they're easy to build and easy to consume.
- What benefits? The web and its core protocol, HTTP, provide a stack of features:
  - Suitable actions (GET, POST, PUT, DELETE, …)
  - Caching
  - Redirection and forwarding
  - Security (encryption and authentication)
- REST is not a standard but an approach, a style, a set of constraints on your architecture that can help you build web-scale systems.

To wrap your repository with a web layer, you must turn to Spring MVC

- **@RestController** indicates that the data returned by each method will be written straight into the response body instead of rendering a template.
- The **@RequestBody** annotation is used to define the request body content type.
- The **@PathVariable** annotation is used to define the custom or dynamic request URI. The Path variable in request URI is defined as curly braces {}
- The **@RequestParam** annotation is used to read the request parameters from the Request URL. By default, it is a required parameter. We can also set default value for request parameters
- We have routes for each operations (**@GetMapping**, **@PostMapping**, **@PutMapping** and **@DeleteMapping**, corresponding to HTTP GET, POST, PUT, and DELETE calls).
- *ResponseEntity represents the whole HTTP response: status code, headers, and body. As a result, we can use it to fully configure the HTTP response.*

- The default **HTTP** request method is **GET**. This method does not require any Request Body. You can send request parameters and path variables to define the custom or dynamic URL.
- The **HTTP POST** request is used to create a resource. This method contains the Request Body. We can send request parameters and path variables to define the custom or dynamic URL.
- The **HTTP PUT** request is used to update the existing resource. This method contains a Request Body. We can send request parameters and path variables to define the custom or dynamic URL.
- The **HTTP Delete** request is used to delete the existing resource. This method does not contain any Request Body. We can send request parameters and path variables to define the custom or dynamic URL.

Spring

Naming Convention for REST API

Resources/Endpoints

1. Use nouns to represent resources and not verbs (/folders - ~~/createFolder~~)
2. Use plural resources (/users/21 - ~~/user/21~~)
3. Use lower-case

Resources	GET read	POST create	PUT update	DELETE
/books	Return a book list	Create a new book	Update all books	Delete all book
/books/145	Return a single book	not allowed (405)	Update only a book	Delete only a book

4. Use hyphens (spinal case) to improve readability of URIs. (/users/noam/reset-password - ~~/users/noam/resetPassword)~~
5. uses the PUT, POST and DELETE methods to alter the state of a resource. Do not use the GET method for state changes

6. Use the sub-resources to describe the relationships
   (GET /books/411/authors/1 → Returns the author #1 of the book411)

7. Implements filtering, sorting, selecting specific fields and paging for collections:
   GET /books?author=Franz+Kafka → Returns The list of books written by Kafka
   GET /books?pages<=200 Ritorna a list of books that have a maximum of 200 pages
8. Allow sorting based on one or more fields:
   GET /books?sort=-pages,+author
9. Selection of fields:
   GET /books?fields=title,author,id
10. Handle errors using HTTP status codes:

200 - OK - All right
201 - OK - A new resource has been created
204 - OK - The resource was successfully deleted
304 - Not modified - The data has not changed. The customer can use the cached data
400 - Bad Request - Invalid request. The exact error should be explained in the error payload (which we will discuss shortly). For example. "The JSON is invalid"
401 - Unauthorized - The request requires user authentication
403 - Forbidden - The server has understood the request, but according to the rights of the applicant, access is not allowed.
404 - Not Found - There is no resource behind the requested URI.
422 - Unprocessable Entity - must be used if the server cannot process the entity, for example if an image cannot be formatted or required fields are missing in the payload.
500 - Internal Server Error - API developers should avoid this error. If a global application error occurs, the stacktrace must be logged and not sent in the response to the user.

- @ResponseBody signals that this advice is rendered straight into the response body.
- @ExceptionHandler configures the advice to only respond if an EmployeeNotFoundException is thrown.
- @ResponseStatus says to issue an HttpStatus.NOT_FOUND, i.e. an HTTP 404.
- The body of the advice generates the content. In this case, it gives the message of the exception.