kirsch & kern
IT Experts and Solutions

GERMAN STANDARD

# Testproject

## Let´s do a test project. We are very curious to see your results.

| | |
|---|---|
| **Skill(s) to be tested** | • Java, senior level |
| **Time to finish it** | • Please do it within about 8 hours |

**Technical requirements**

You should develop a microservice from scratch. The application should use Spring Boot 2 as Framework and provide a REST API and use a relational memory database as persistence unit. You have to create the data model by analysing the given business requirements below.

The API should be intuitive to use and should be well documented. The application has to be tested fully automatically so that no manual driven test cases are needed before deploying to production environment. You can neglect testing the integration with other microservices. The application should be self-contained, buildable on a third-party computer and fully production ready.

As build tool you should use maven. Consider that another person takes care of the product in future, so a high code quality and common used libraries are mandatory. Create a local git repository and commit to it periodically. If you have finished with development please push all your changes to the git repository, export the repository to a zip file with the usage of *git archive* and send us the result.

**Business requirements**
The business context is located in the financial sector. The application should realize following business cases. You should implement as much cases as possible but considering the cases are prioritised from more important to less important.

- Create a customer with full name, day of birth, address and a rating class which defaults to ‚2‘.
- Query names and addresses of all customers by their last name and allow to sort the result.
- Transfer money from one account to another resulting in a posting.
- Create a new account for a given customer.
- Create a new credit for a given customer.
- List all accounts of one customer with their current balance.
- List all credits of one customer with their original term, remaining term, original credit amount and the current credit amount.
- List all postings of the financial institution for a given booking date.
- List all postings with the account id and customer name of source and destination of one customer and make the result sortable and page-able.
- Payoff a part of a credit by transferring money from an account.
- The application should be secured with a login backed with only one hardcoded user and password combination. If a user is not logged he should not have access to the system.
- Show the balance for one customers.

- Show the balance for the financial institution.
- The API should have an online documentation.
- The first booking should be configurable in the properties of the application.
- Process to the next booking date. A booking date should be every weekday regardless of holidays.
- List all credits with original credit amount, current credit amount and customer name which are exceeded their original terms.
- If a customer paid off a credit he will be awarded to a better rating class but at maximum to ‚1'.
- If a customer didn't paid off a credit before the remaining term is below zero his rating class will be set to ‚4'.
- List all customers with name grouped by their current rating class.