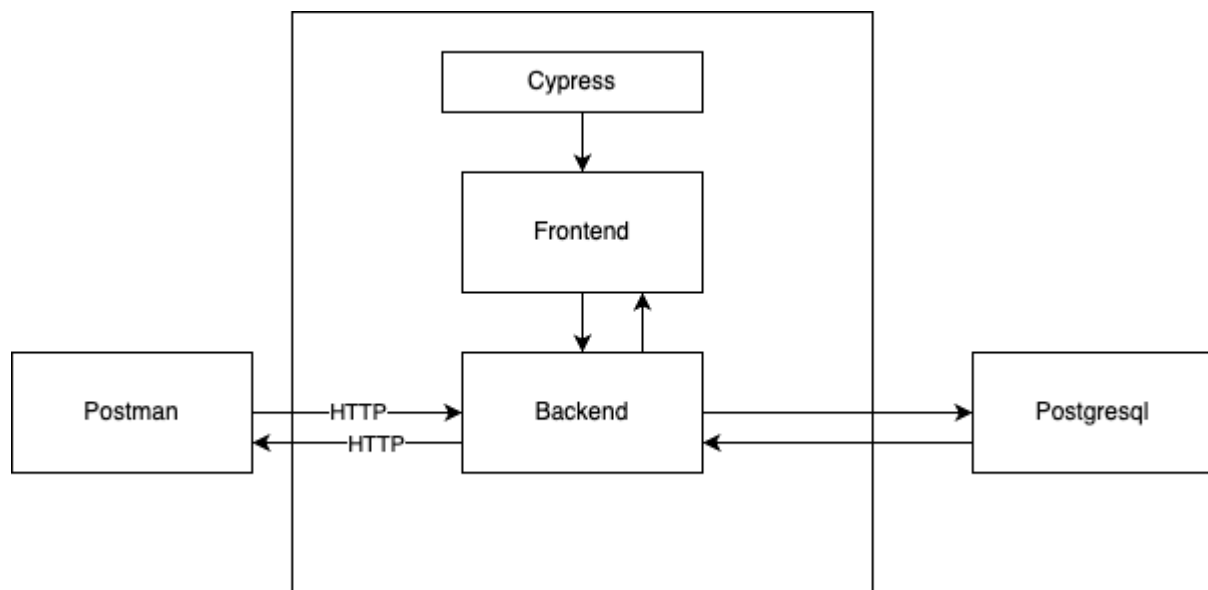


Seda Deniz Özdoğar
Gülşah Tiye

e180502006@stud.tau.edu.tr
e170502003@stud.tau.edu.tr

1. SYSTEMÜBERBLICK

Wir planen, die von uns erstellten Dienste auf zwei verschiedene Arten zu testen. Im Rahmen des Integrationstests sollten Restful-Dienste getestet werden. Aus diesem Grund werden HTTP-Anfragen modellgemäß über Postman versendet und Dienste getestet. Für End-to-End-Tests wird Cypress verwendet. Mit Cypress ist es möglich, sowohl UI-Tests als auch Komponententests durchzuführen.








2. SYSTEMTESTFÄLLE

Das System kann über eine Benutzeroberfläche oder über Postman getestet werden. Beim Testen mit Postman werden Endpunkt- und Methodeninformationen für den entsprechenden Dienst benötigt. Die Endpunkt- und Methodeninformationen werden in Postman zur Anfrage hinzugefügt. Im Body-Bereich werden Informationen eingegeben, dass der Content gesendet wird, und dann wird eine POST-Anfrage gesendet. Es wird erwartet, dass die Antwort den Statuscode 401 enthält.


| System Test Case ID Name | Unbefugter Benutzer kann Todo nicht hinzufügen |
|--------------------------|--|
| Preconditions | <ul style="list-style-type: none"> - User should open the app. - User should not be logged in. - Server calculates the session id. - GetSessionId return session id to client - AddTodo(content:string) works |
| Test Steps | <ul style="list-style-type: none"> - User open the page. - User types a todo. - User clicks the submit button. - Service returns 401 |
| Post Condition | none |
| Post Condition | Nothing is written to the database |
| Test Data | This is a todo request coming from an unauthorized user! |
| Expected Result | Service returns 401 |
| Actual Result | Service returns 401 |
| Verdict (Pass/Fail) | Passed |


Testergebnis:



 http://localhost:3000/api/controller/todo  Save  

POST 





http://localhost:3000/api/controller/todo





Send 

Params Auth Headers (22) Body  Pre-req. Tests Settings Cookies Beautify

raw  JSON 

```
1 { "content": "This is a todo request coming from not authorized user!" }
```

Body   401 Unauthorized 30 ms 280 B  Save as Example 

Pretty Raw Preview Visualize JSON    

```
1 {  
2   "message": "Please login to start creating todos!"  
3 }
```

Um die Datenbankverbindung über Unittests zu testen, werden sowohl der Service als auch die Datenbank zuerst gemockt. Dadurch wird eine Testumgebung erstellt. Anschließend wird eine Funktion erstellt, die über den Service eine Testanfrage an die Datenbank sendet. Für diese Anfrage wird der Test entsprechend der erwarteten Bedingung durchgeführt.

| Compenent Integration Test Case ID Name | Neue Aufgaben in die Datenbank schreiben |
|--|---|
| Preconditions | <ul style="list-style-type: none"> - User should login. - getServerSession(id) should generate session id in sessions table. - Service should be mocked. - Generate test database |
| Test Steps | <ul style="list-style-type: none"> - Mocking the service - Cypress calls addTodo(content:string) service |
| Post Condition | none |
| Test Data | {"content": "This todo is for testing purpose"} |
| Expected Result | There is a new todo in Todo table with specified todo.content |
| Actual Result | There is a new todo in Todo table with specified todo.content |
| Verdict (Pass/Fail) | Passed |

3. RÜCKVERFOLGBARKEIT

Wenn das System manuell getestet wird, wird zuerst über die Benutzeroberfläche eine Anmeldung gemäß den Preconditions durchgeführt. Anschließend werden über das Formular To-Do-Informationen hinzugefügt. Nach dem Drücken der Submit-Schaltfläche gelangt die entsprechende Anfrage zum To-Do-Controller-Service. Dort werden verschiedene Kontrollen durchgeführt. Um festzustellen, ob der entsprechende Benutzer angemeldet ist oder nicht, wird die Sitzungsinformation abgerufen und anschließend werden die erforderlichen Kontrollen durchgeführt. Wenn die Kontrollen erfolgreich sind, wird die Anfrage an den ORM (Object-Relational Mapping) weitergeleitet. Das ORM erstellt die erforderliche SQL-Abfrage, um das To-Do in die Datenbank zu schreiben. Wenn der Schreibvorgang in die Datenbank erfolgreich ist, wird der HTTP-Statuscode 200 zurückgegeben.

