



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра автоматики та управління в технічних системах

Лабораторна робота №9  
**ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ ПРОГРАМНОГО  
ЗАБЕЗПЕЧЕННЯ**

Виконав  
студент групи ІА–13:  
Лапушенко А.К.

Київ 2023

**Тема:** Вид взаємодії клієнт-сервер “*Client-server*”.

**Варіант:** POS(Point-of-sales)-system.

### Хід роботи

Клієнт-серверні додатки являють собою найпростіший варіант розподілених додатків, де виділяється два види додатків: клієнти (представляють додаток користувачеві) і сервери (використовується для зберігання і обробки даних). Клієнтська частина отримує дані з сервера, який взаємодіє з базою даних та віддає потрібні дані.

В даній роботі клієнтський додаток був реалізований на ReactJs Axios, а серверний на – Java. Дані передаються у виді JSON.

#### Серверна частина

```
@RestController
@RequiredArgsConstructor
public class UserController {
    private final UserService userService;

    @CrossOrigin(origins = "http://localhost:3000")
    @GetMapping("/users")
    public List<User> getUsers(){
        return userService.listAll();
    }
}
```

#### Клієнтська частина

```
export default class UserComponent extends React.Component {}

constructor(props) {
    super(props)

    this.state = {
        users: []
    };
}

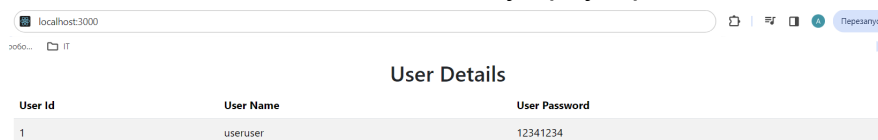
componentDidMount(){
    ApiService.getUsers().then((res) => {
        this.setState({ users: res.data })
    })
    .catch(function (ex) {
        console.log('Response parsing failed. Error: ', ex);
    });
}
```

```

render() {
  return (
    <div>
      <h2 className="text-center">User Details</h2>
      <table className="table table-striped">
        <thead>
          <tr>
            <th>User Id</th>
            <th>User Name</th>
            <th>User Password</th>
          </tr>
        </thead>
        <tbody>
          {
            Array.isArray(this.state.users)?
            this.state.users.map(user =>
              <tr key={user.id}>
                <td>{user.id}</td>
                <td>{user.name}</td>
                <td>{user.password}</td>
              </tr>
            ):null
          }
        </tbody>
      </table>
    </div>
  )
}

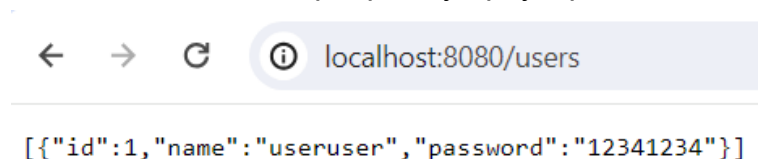
```

Вид в клієнтському браузері



| User Id | User Name | User Password |
|---------|-----------|---------------|
| 1       | useruser  | 12341234      |

Вид в серверному браузері



```
[{"id":1,"name":"useruser","password":"12341234"}]
```

## Висновок

В цій роботі я створив клієнт-серверну архітектуру застосунку, чим збільшив швидкість для користувача та розділив бізнес логіку та представлення для користувача.

## Контрольні питання

### 1. Що таке SOA?

Сервіс-орієнтована архітектура (SOA, англ. service-oriented architecture) — модульний підхід до розробки програмного забезпечення, заснований на використанні розподілених, слабо пов'язаних (англ. Loose coupling) замінних компонентів, оснащених стандартизованими інтерфейсами для взаємодії за стандартизованими протоколами.

### 2. Якими принципами керується SOA?

- Архітектура не прив'язана до певної технології.
- Незалежність організації системи від мов програмування
- Використання сервісів, незалежних від конкретних програм, з одноманітними інтерфейсами доступу до них
- Організація сервісів як слабозв'язаних компонентів для побудови систем