

GO + WebService

Krzysztof Anioł

Jakub Janecki

Karol Kiełbasa

Monika Mazur

Analiza dostępności wybranej technologii

- W języku programowania Go występuje mnogość framework'ów, które mogą posłużyć do budowy aplikacji wspierającej API REST'owe. W celu wybrania najbardziej odpowiedniego dla naszego projektu, przeszukaliśmy książki [9] czy porównania [11]. Mając na uwadze powyższe informacje, zdecydowaliśmy się na Beego.
- Beego wspiera REST oraz model MVC, monitoruje zużycie CPU oraz pamięci. Beego sam w sobie zawiera mnóstwo bibliotek, dzięki czemu nie ma potrzeby szukania oraz dodawania ich do projektu. Wbudowane moduły wspierają kontrolę sesji, cache czy ORM. Beego posiada kilka tutoriali wprowadzających w jego używanie [3].

Użyte technologie

- Oczywiście – język Go
- Framework Beego
- REST
- Integracje z zewnętrznym oprogramowaniem:
 - AWS S3
 - Draftable Comparison API
 - CloudConvert API
 - Java
 - Python

Narzędzia komunikacyjne

- Skype
- GitHub
- Trello



Metodyka pracy

The screenshot displays a Kanban board interface for 'SRiR Go + Webservice'. The board is organized into three main columns: 'To Do', 'Doing', and 'Done'. Each column contains several task cards, each with a title, a progress bar, a priority indicator (bell icon), a due date, and assigned team members (represented by initials in circles).

Header: SRiR Go + Webservice | SRiR projekt Free | Team Visible | MM JJ K KA 4 | Share

To Do Column:

- Prezentacja (Progress: 25%, Priority: 1, Due: Jan 13, Assigned: MM)
- Dokumentacja kodu - Doxygen (Progress: 25%, Priority: 1, Due: Jan 13, Assigned: JJ, K, MM)
- Czy implementacja działa na 4+ komputerach? (Progress: 10%, Priority: 1, Due: Jan 13, Assigned: JJ, K, MM)

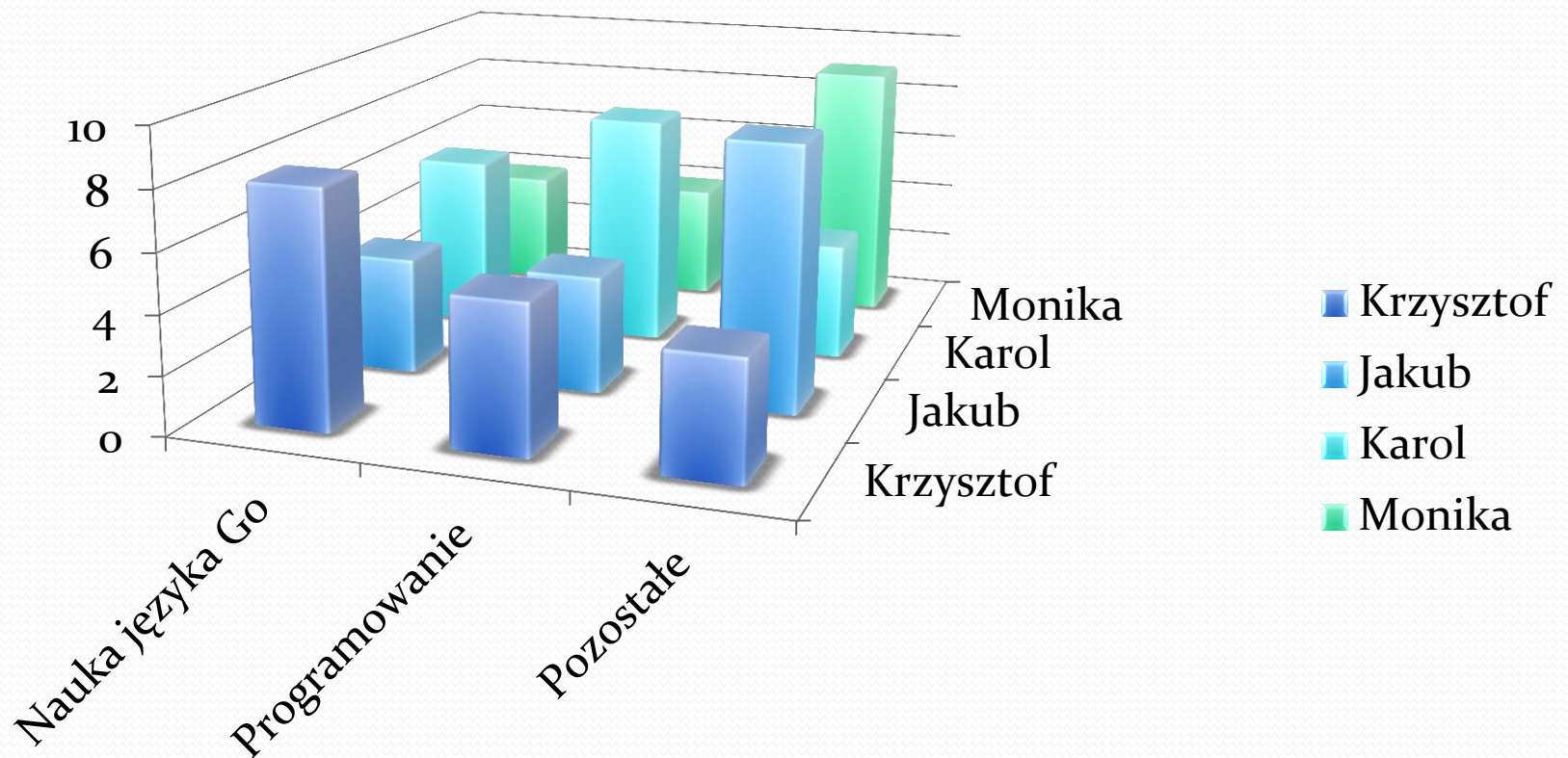
Doing Column:

- Diagramy UML (Progress: 50%, Priority: 1, Due: Jan 13, Assigned: JJ)
- Testy jednostkowe - KAŻDY (Progress: 25%, Priority: 1, Due: Jan 13, Assigned: JJ, K, MM)
- Po 1 unikalnym przypadku testowym całego opracowanego systemu. - KAŻDY (Progress: 25%, Priority: 1, Due: Jan 13, Assigned: JJ, K, MM)
- Testy całego swojego modułu - KAŻDY (Progress: 10%, Priority: 1, Due: Jan 13, Assigned: JJ, K, MM)
- Analiza - szybkość działania (Progress: 10%, Priority: 1, Due: Jan 13, Assigned: JJ, MM)

Done Column:

- Implementacja - OS 2 (Progress: 100%, Priority: 2, Due: Dec 28, 2018, Assigned: K, MM)
- Implementacja - OS 3 (Progress: 100%, Priority: 1, Due: Dec 28, 2018, Assigned: JJ, K, MM)
- Implementacja - OS 1 (Progress: 100%, Priority: 1, Due: Dec 28, 2018, Assigned: K)
- Analiza dostępności wybranej technologii (Progress: 25%, Priority: 1, Due: Dec 28, 2018, Assigned: MM)
- Analiza porównawcza implementacji - nakład pracy (Progress: 25%, Priority: 1, Due: Dec 28, 2018, Assigned: MM)
- Bibliografia (Progress: 25%, Priority: 1, Due: Dec 28, 2018, Assigned: MM)

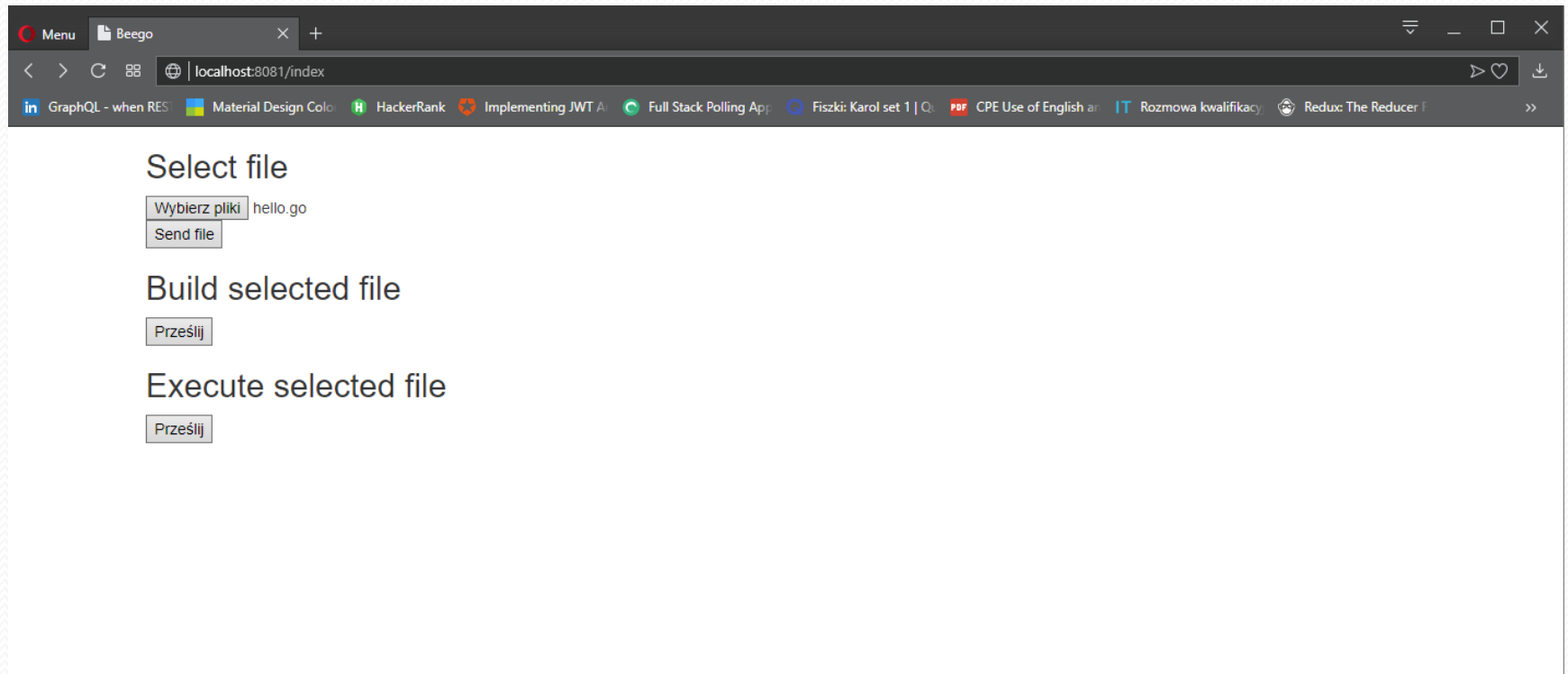
Analiza nakładu czasu



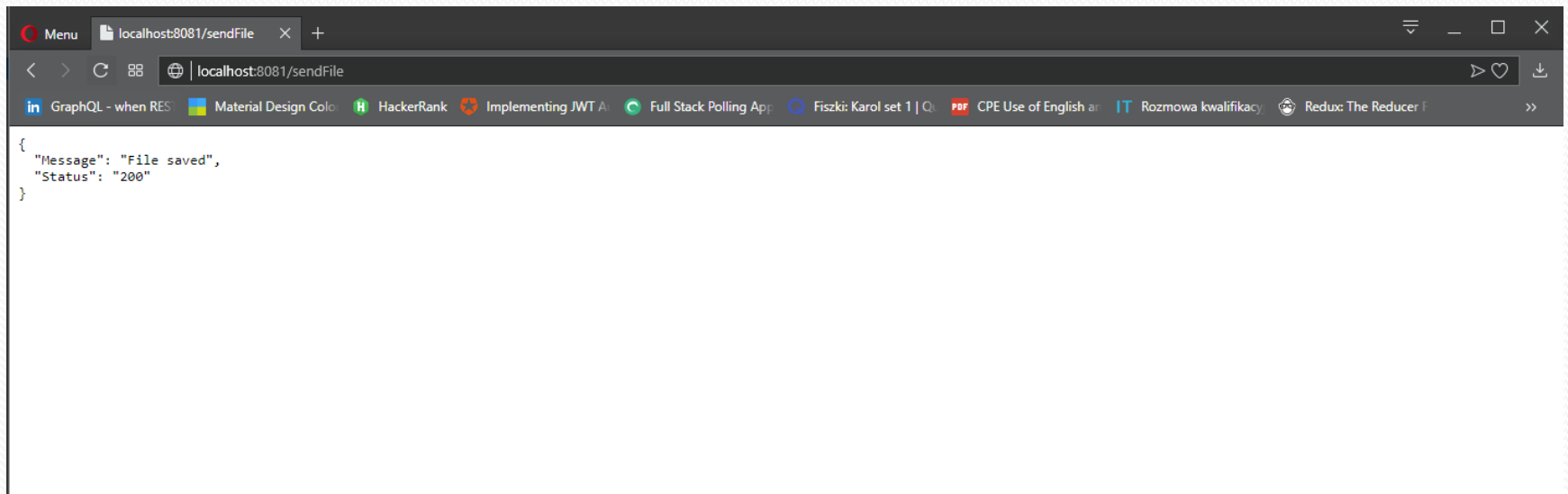
Kod źródłowy

<https://github.com/karol-kielbasa/SRiR>

GUI Klienta



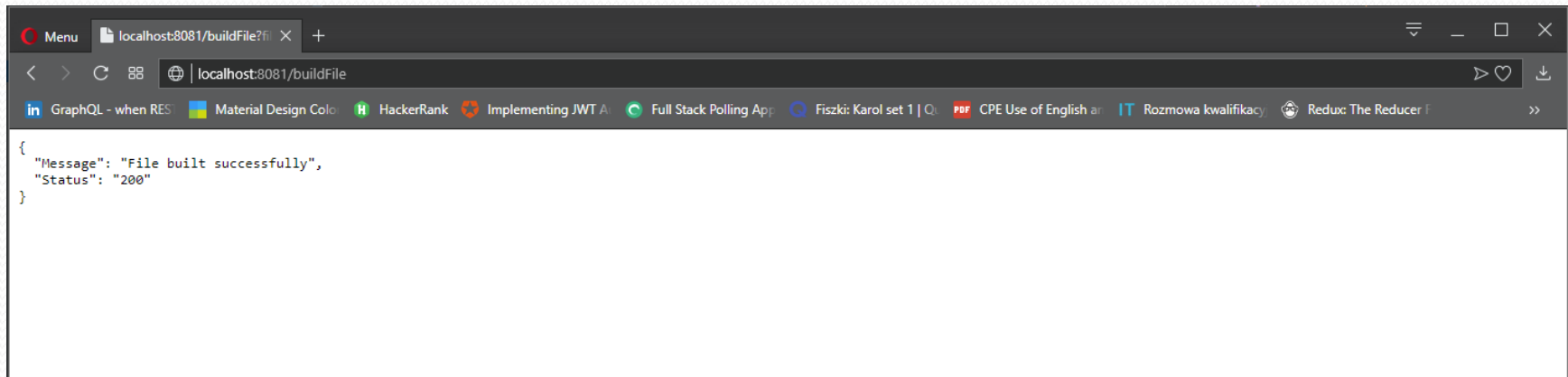
Odpowiedź serwera po wysłaniu prawidłowego pliku



A screenshot of a web browser window. The address bar shows 'localhost:8081/sendFile'. The page content displays a JSON response from the server, indicating a successful file upload. The response is: { "Message": "File saved", "Status": "200" }. The browser's tab bar shows multiple open tabs, including 'Menu', 'localhost:8081/sendFile', and various development-related links like 'GraphQL - when RES', 'Material Design Colo', 'HackerRank', 'Implementing JWT A', 'Full Stack Polling App', 'Fiszki: Karol set 1 | Q', 'CPE Use of English an', 'Rozmowa kwalifikacy', and 'Redux: The Reducer F'.

```
{
  "Message": "File saved",
  "Status": "200"
}
```

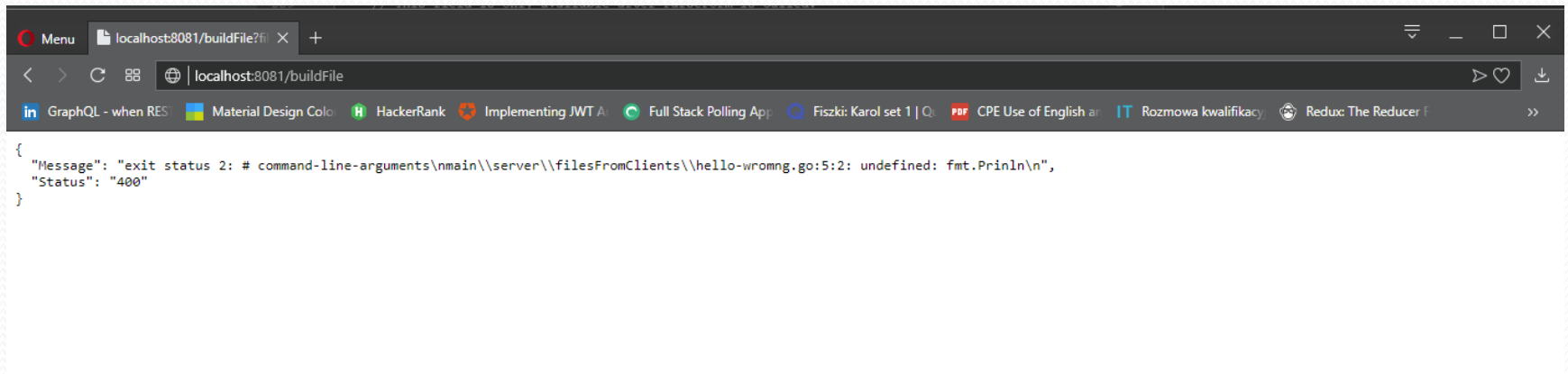
Odpowiedź serwera po poprawnym zbudowaniu pliku



A screenshot of a web browser window. The address bar shows the URL `localhost:8081/buildFile`. The browser's tab bar includes a 'Menu' tab and several open tabs with titles like 'GraphQL - when RES', 'Material Design Col...', 'HackerRank', 'Implementing JWT A...', 'Full Stack Polling App', 'Fiszki: Karol set 1 | Q...', 'CPE Use of English an...', 'Rozmowa kwalifikacy...', and 'Redux: The Reducer f...'. The main content area of the browser displays a JSON response:

```
{
  "Message": "File built successfully",
  "Status": "200"
}
```

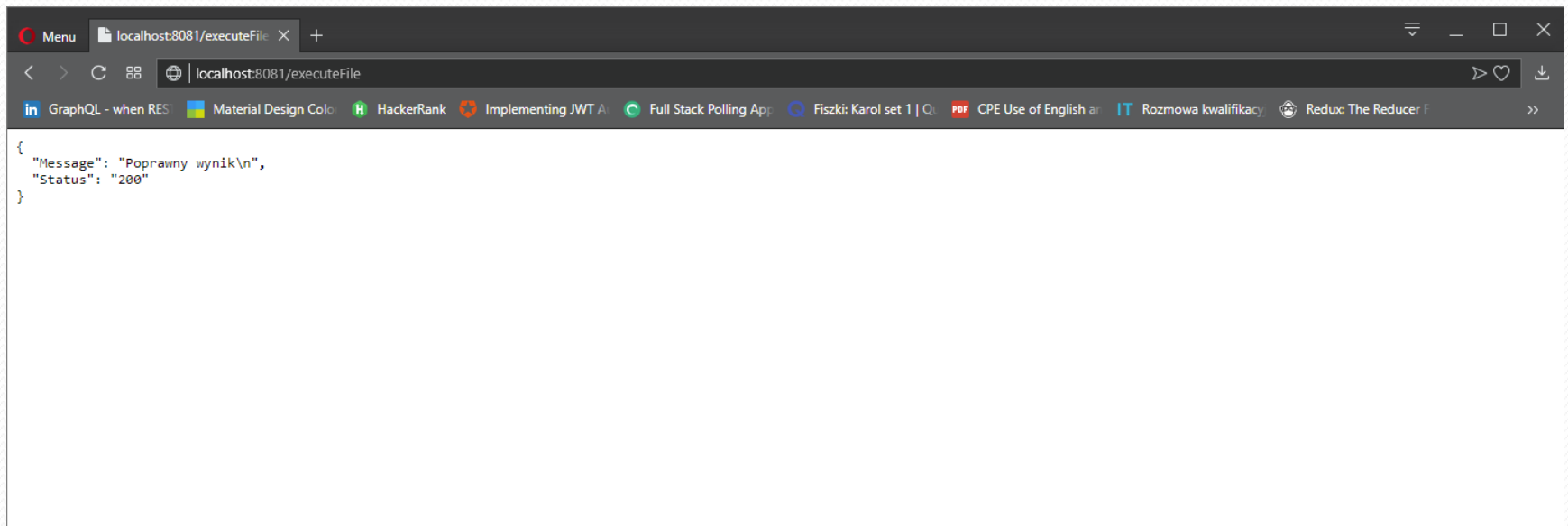
Odpowiedź serwera przy błędzie budowania pliku



The screenshot shows a web browser window with the address bar displaying `localhost:8081/buildFile`. The browser's developer tools are open, showing a JSON response from the server. The response is a JSON object with two properties: `"Message"` and `"Status"`. The `"Message"` property contains the text `"exit status 2: # command-line-arguments\nmain\\server\\filesFromClients\\hello-wromng.go:5:2: undefined: fmt.Println\n"`, and the `"Status"` property contains the text `"400"`. The browser's tab bar shows several open tabs, including `GraphQL - when RES`, `Material Design Colo`, `HackerRank`, `Implementing JWT A`, `Full Stack Polling App`, `Fiszki: Karol set 1 | Q`, `CPE Use of English ar`, `Rozmowa kwalifikacy`, and `Redux: The Reducer F`.

```
{
  "Message": "exit status 2: # command-line-arguments\nmain\\server\\filesFromClients\\hello-wromng.go:5:2: undefined: fmt.Println\n",
  "Status": "400"
}
```

Odpowiedź serwera po wykonaniu programu



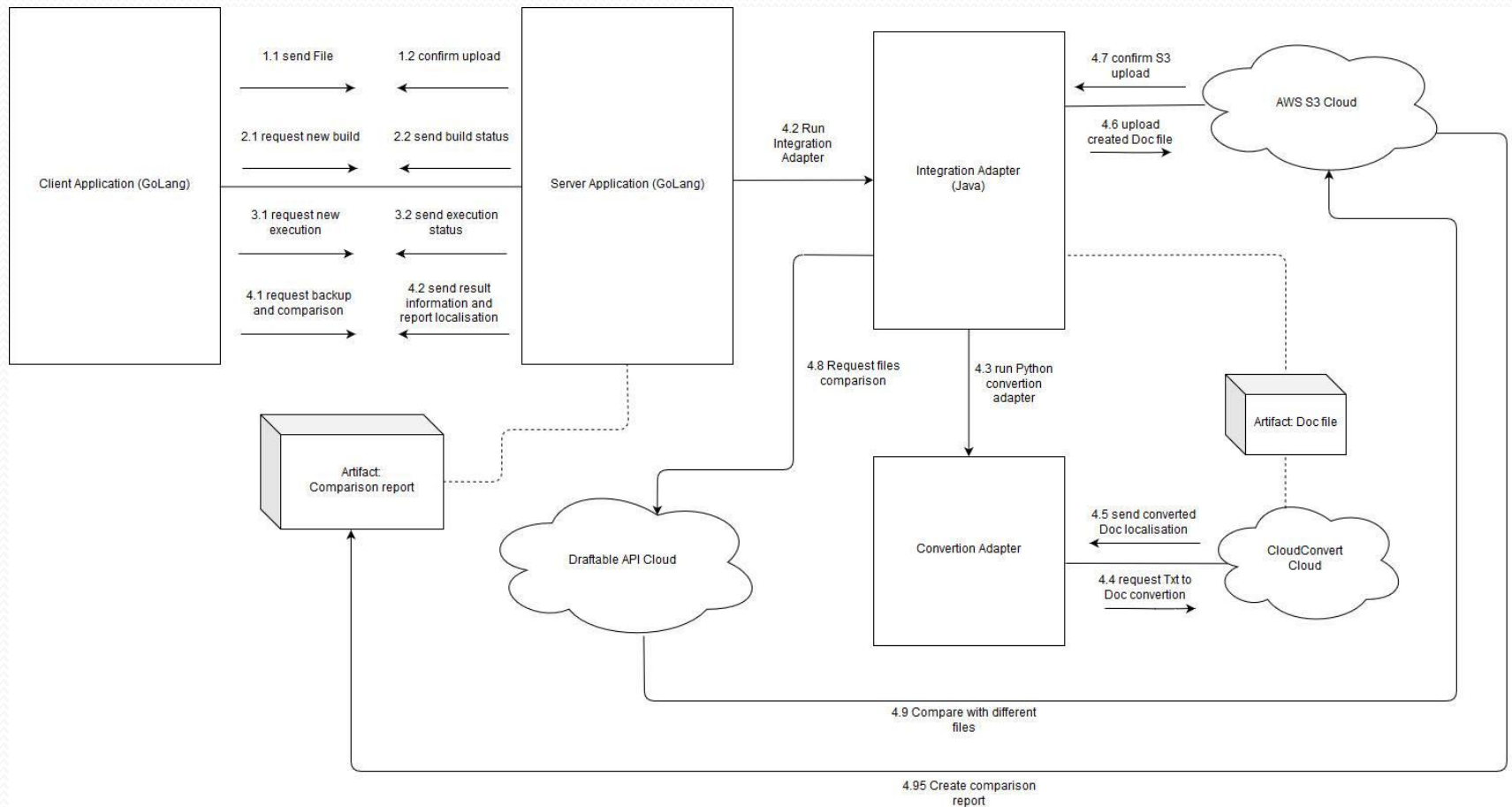
A screenshot of a web browser window. The address bar shows the URL `localhost:8081/executeFile`. The browser's tab bar includes a 'Menu' tab and several other tabs with various titles. The main content area of the browser displays a JSON object representing the server's response.

```
{  
  "Message": "Poprawny wynik\n",  
  "Status": "200"  
}
```

Dokumentacja

- Dokumentacja naszego projektu została wygenerowana przez GoDoc
- <https://godoc.org/github.com/karol-kielbasa/SRiR/main>

Diagram komunikacji



Bibliografia

- <https://golang.org/doc/> Oficjalna strona języka GO, uzyskano dostęp: 18-01-2019
- <https://www.pluralsight.com/courses/go> Kurs dotyczący języka Go w serwisie Pluralsight, uzyskano dostęp: 18-01-2019
- <https://www.pluralsight.com/courses/beego-go-web-app-framework> Kurs dotyczący frameworka Beego w serwisie Pluralsight, uzyskano dostęp: 18-01-2019
- <https://beego.me/> Oficjalna strona frameworka Beego, zawierająca dokumentację, uzyskano dostęp: 18-01-2019
- Varghese, S.: Go Recipes: A Problem-Solution Approach, wydanie 1, Apress 2016, ISBN: 978-1-4842-1189. Rozdziały: 1. Beginning Go, 2. Go Fundamentals
- <https://github.com/shijuvar/go-recipes> Przykłady, kod źródłowy opisany w książce: Go Recipes: A Problem-Solution Approach, uzyskano dostęp: 18-01-2019
- Newmarch, J.: Network Programming with Go, wydanie 1, Apress 2017, ISBN: 978-1-4842-2692-6. Rozdziały 2. Overview of the Go Language, 14. REST
- <https://github.com/tumregels/Network-Programming-with-Go> Przykłady, kod źródłowy opisany w książce: Network Programming with Go, uzyskano dostęp: 18-01-2019
- Varghese, S.: Web Development with Go Building Scalable Web Apps and RESTful Services, wydanie 1, Apress 2015, ISBN: 978-1-4842-1053-6. Rozdziały 4. Getting started with Web Development, 9. Building RESTful Services.
- <https://github.com/apress/web-dev-w-go> kod źródłowy opisany w książce Web Development with Go Building Scalable Web Apps and RESTful Services, uzyskano dostęp: 18-01-2019
- <https://github.com/diyan/go-web-framework-comparision> porównanie frameworków w języku Go, uzyskano dostęp: 18-01-2019
- Gulabani, S.: Practical Amazon EC2, SQS, Kinesis, and S3 A Hands-On Approach to AWS, wydanie 1, Apress 2017, ISBN: 978-1-4842-2841-8. Rozdział 5: Hands-on Simple Storage Services (S3)
- <https://draftable.com/comparison-api> Oficjalna strona Draftwable Comparison API, zawierająca dokumentację oraz przykłady, uzyskano dostęp: 18-01-2019
- <https://cloudconvert.com/api> Oficjalna strona CloudConvert API, zawierająca dokumentację, uzyskano dostęp: 18-01-2019

Dziękujemy za uwagę!

