

## Analiza dostępności technologii REST w środowisku Ruby

Użyta w naszym projekcie technologia to REST( Representational State Transfer) czyli pewien wzorzec tworzenia systemów rozproszonych. Opiera się on na adresach URL. To przez nie wykonujemy wszystkie działania na poszczególnych obiektach, dzięki odpowiednim metodom. Wyróżniamy 4 najważniejsze metody tego protokołu :

- GET (pobieranie danych)
- POST (dodawanie danych)
- PUT (modyfikacja danych)
- DELETE (usuwanie danych)

REST posiada swoje implementację w wielu językach programowania, między innymi w języku Ruby.

W ramach naszego projektu do implementacji tej technologii użyto biblioteki *Sinatra* . Który pozwala na wykonanie komunikacji w konwencji REST.

Poniżej przedstawiono najpopularniejsze rozwiązania:

### 1. Sinatra

Sinatra cechuje się zwięzłą składnią i prostotą implementacji. Napisanie aplikacji typu 'Hello world' mieści się w 4 bardzo krótkich liniach:

```
require 'sinatra'
get '/frank-says' do
  'Put this in your pipe & smoke it!'
end
```

Co więcej, Sinatra posiada bardzo dobrą dokumentację i jest projektem typu Open-source.

## 2. Ruby Grape

Kolejnym możliwym rozwiązaniem pozwalającym na budowanie aplikacji w stylu REST jest framework 'Ruby Grape'. Który był odpowiedzią na większe i bardziej skomplikowane narzędzia np. Sinatra czy Rails.

Aby móc z niego korzystać należy dołożyć do projektu gem poprzez komendę `'gem install grape'`. Dodatkowo, można go łączyć z innymi lekkimi frameworkami np. z Sinatrą.

Przykładowy kod wykorzystujący metodę POST :



```
## This takes care of parameter validation
params do
  requires :name, type: String
  requires :address, type:String
  requires :age, type:Integer
end
## This takes care of creating employee
post do
  EmpData.create!({
    name:params[:name],
    address:params[:address],
    age:params[:age]
  })
end
```

### 3. Cuba

Cuba to kolejna propozycja, która pozwala na korzystanie z wzorca REST. Instalacja przebiega w taki sam sposób jak rozwiązania powyżej. Cuba posiada całkiem obszerną dokumentację znajdującą się w serwisie Github.



Przykładowy kod z użytą metodą GET

```
# cat hello_world.rb
require "cuba"
require "cuba/safe"

Cuba.use Rack::Session::Cookie, :secret => "__a_very_long_string__"

Cuba.plugin Cuba::Safe

Cuba.define do
  on get do
    on "hello" do
      res.write "Hello world!"
    end

    on root do
      res.redirect "/hello"
    end
  end
end
```

#### Podsumowanie:

Technologia REST w języku Ruby jest popularna. Do dyspozycji jest kilka gotowych bibliotek/frameworków. Powyżej przedstawiono 3 najpopularniejsze. Wszystkie posiadają dobre dokumentację i są proste w implementacji.