

HealthMatch Interview: TreeMatch

TreeMatch is a service that tells you what kind of tree you should plant in your garden! It loads a collection of questions from a database (well... a JSON file) and guides users through the questionnaire, before matching them with a tree species.

Your task is to build the API server for TreeMatch! Send us an archive or (private) version control repository invite with:

- Your code, in any statically-typed language (Kotlin is a bonus!)
- A README with instructions for compiling and running your code

We don't want you to have to spend all day on this, so show us your knowledge of best practices but don't feel the need to include extra features. Feel free to include any quick notes on what you'd do differently on a real/production codebase.

API

Requests and responses are both in JSON. The API has two methods:

- `/api/begin` - Returns the first question
- `/api/answer` - Receives a POSTed answer, and returns the next question (or a tree match!)

`/api/begin`

Returns a question, of the form:

```
{
  "question": {
    "step_id": 1,
    "question": "What's your favourite kind of tree?",
    "answers": ["fruit trees", "gum trees", "red-black trees"]
  }
}
```

- `"step_id"` is the ID of the current step in the quiz
- `"question"` is the text of a question
- `"answers"` is a list of strings, the answers the user may choose from

`/api/answer`

Accepts POSTs with a body like:

```
{
  "step_id": 1,
  "answer": "gum trees"
}
```

- `"step_id"` is the numerical ID of the step that is being answered
- `"answer"` is a string, the answer

Responds with either another question (the same format as `/api/begin`) or a match:

```
{
  "match": {
    "name": "Eucalyptus tree",
    "description": "This tree is an Aussie bush favourite!"
  }
}
```

Data source

All the data for our quiz is loaded from a file, `questions.json`, which we provide. It contains three things:

- `"questions"` - A list of all possible questions, and their valid answers
- `"results"` - A list of all possible trees that can be matched with
- `"steps"` - A list of quiz steps. A step is a question, along with a mapping from answer to next question.

Note that the same question may be used multiple times. And the same tree result might be possible from multiple different answers.

Here is a simple valid `"questions.json"`:

```
{
  "questions": [
    {
      "id": 1,
      "question": "Do you like fruit?",
      "validation": ["yes", "no"]
    }
  ],
  "results": [
    {
      "id": 1,
      "name": "Apple tree",
      "description": "The most basic fruit tree"
    },
    {
      "id": 2,
      "name": "Willow tree",
      "description": "This tree is mysterious and cool"
    }
  ],
  "steps": [
    { "id": 1, "question_id": 1, "answers": { "yes": 2, "no": 3 } },
  ]
}
```

```
{ "id": 2, "result_id": 1 },  
{ "id": 3, "result_id": 2 }  
]  
}
```

Make sure you validate "[questions.json](#)" when you load it! We don't want any malformed or broken quizzes slipping through!