



Implement an API for our Bots. This should consist of a HTTP/REST backend and a database. The backend will have two resources:

**/bots:** the bot resource represent the bots registered in our Bot platform. The /bots endpoint have to manage all operations related to bots i.e. create, read, update and delete.

e.g.:

- POST /bots

```
{  
  "id": "36b9f842-ee97-11e8-9443-0242ac120002",  
  "name": "Aureo"  
}
```

- GET /bots/:id

```
{  
  "id": "36b9f842-ee97-11e8-9443-0242ac120002",  
  "name": "Aureo"  
}
```

**/messages:** a bot exchange messages with users. All users are considered anonymous in our platform -- hence they don't need to be registered nor logged in. Nevertheless, users' session are represented by an unique id. It can be either the from or the to attribute in a message, depending on whether he received or sent it.

All the messages exchanged between a bot and an user, during a conversation, will be correlated by a conversationId. The API should be capable of (i) registering new messages, for a given conversation; (ii) return a message by its id; and (iii) return all messages of a given conversation. Messages won't be updated nor deleted.

e.g.:

- POST /messages

```
{  
  "conversationId": "7665ada8-3448-4acd-a1b7-d688e68fe9a1",  
  "timestamp": "2018-11-16T23:30:52.6917722Z",  
  "from": "36b9f842-ee97-11e8-9443-0242ac120002",  
  "to": "16edd3b3-3f75-40df-af07-2a3813a79ce9",  
  "text": "Oi! Como posso te ajudar?"  
}
```

- GET /messages/:id

```
{  
  "id": "16edd3b3-3f75-40df-af07-2a3813a79ce9",  
  "conversationId": "7665ada8-3448-4acd-a1b7-d688e68fe9a1",  
  "timestamp": "2018-11-16T23:30:52.6917722Z",  
  "from": "36b9f842-ee97-11e8-9443-0242ac120002",  
  "to": "16edd3b3-3f75-40df-af07-2a3813a79ce9",  
  "text": "Oi! Como posso te ajudar?"  
}
```

- GET /messages?conversationId=:conversationId

```
[  
  {  
    "id": "16edd3b3-3f75-40df-af07-2a3813a79ce9",  
    "conversationId": "7665ada8-3448-4acd-a1b7-d688e68fe9a1",  
    "timestamp": "2018-11-16T23:30:52.6917722Z",  
    "from": "36b9f842-ee97-11e8-9443-0242ac120002",  
    "to": "16edd3b3-3f75-40df-af07-2a3813a79ce9",  
    "text": "Oi! Como posso te ajudar?"  
  },  
  ]
```

```
{  
  "id": "67ade836-ea2e-4992-a7c2-f04b696dc9ff",  
  "conversationId": "7665ada8-3448-4acd-a1b7-d688e68fe9a1",  
  "timestamp": "2018-11-16T23:30:57.5926721Z",  
  "from": "16edd3b3-3f75-40df-af07-2a3813a79ce9",  
  "to": "36b9f842-ee97-11e8-9443-0242ac120002",  
  "text": "Gostaria de saber meu saldo?"  
}  
]
```

### **Implementation**

TODO - fill in with your implementation details here. Feel free to pick and choose any development stack i.e. language, database, test tools, etc. Depict the software engineering details you would address on relation to backend development e.g. build & deployment, scalability, fault tolerance, tests, etc. Please, do so even if you don't have enough time to develop it during the time you're given. Please lay out the steps we need to take in order to run your implementation locally.

Describe any outstanding implementation details, as well as, you may also list any technical debt you have identified on your solution, which was not done due to the time restriction.