Inbenta Developer Challenge

https://www.inbenta.com

1. Disclaimer

THIS DOCUMENT IS AN INTELLECTUAL PROPERTY OF INBENTA TECHNOLOGIES INC., FOR INTERNAL USE AND DISCLOSURE TO SELECTED CUSTOMERS, INVESTORS AND PARTNERS.

IT IS A CONFIDENTIAL WORK PROTECTED BY COPYRIGHT AND TRADE SECRET LAW AND NEITHER IT NOR ANY OF THE INFORMATION CONTAINED THEREIN MAY BE DISCLOSED, USED OR REPRODUCED IN ANY MANNER, OR FOR ANY PURPOSE, EXCEPT BY WRITTEN PERMISSION OF INBENTA TECHNOLOGIES INC.

2. Table of Contents

1. Disclaimer 2

2. Table of Contents 3

3. Introduction 4

4. YodaBot 5

5. Text Analyzer 7

6. API Access and Credentials 9

3. Introduction

The purpose of this document is to describe the challenge that Inbenta has developed to assess your research and development skills as well as for you to have a glimpse on the kind of projects you are your team might be working on.

There are 2 tests in this challenge:

- *YodaBot*: developing the user interface for a chatbot that has learned to speak like Yoda.
- *Text Analyzer*: this is about designing a solution for Text Analytics and Natural Language Processing, you won't need to actually code the final application.

Last chapter contains information about the API and credentials you will need to accomplish this project.

You have 1 week to complete both tests in this challenge since the moment you receive this document. Shall you need more time, please send us an email to challenge@inbenta.com, (Cc. selec.angela@gmail.com), we will try to accommodate your request.

If you have any questions, please don't hesitate to contact us at challenge@inbenta.com. (Cc. selec.angela@gmail.com),

May the Force be with you!

4. YodaBot

The purpose of this test is to evaluate your research & development skills. To this end, you will code a simple integration using PHP, HTML and JavaScript (we encourage you to use Vue.js) that can talk to YodaBot, an intelligent Bot who has learned to talk like Yoda.

This page must be hosted by you and publicly accessible. We recommend using a free account with Heroku, but you can choose the site/platform you prefer. You can see a demo of this page in the video following this link:

> https://www.dropbox.com/s/nk2dbzr0e9xksak/Inbenta_challenge_example.mov?dl=0
>
> *Note: next chapter contains YodaBot API credentials, documentation and some auth pseudo-code.*

In order to complete the test successfully, you must:

1. Display an HTML form with:
   a. An input text that will be used to send messages.
   b. A submit button.
2. When the form is submitted (form cannot be submitted if the message is empty):
   a. Display a "*writing*.." text next to the form.
   b. Display the introduced message as a conversation.
   c. Perform an AJAX request sending the message value.
3. On server-side, process the AJAX request to:
   a. Connect to the Inbenta Chatbot API and:
      1. Open a new conversation if there isn't one already open (use the /conversation endpoint).
      2. Send the message introduced to the current conversation (use the /conversation/message endpoint).
   b. Return the Chatbot API response (from /conversation/message) to the client-side of the AJAX request.
4. Once the AJAX request is completed:
   a. Hide the "writing…" text.
   b. Display the message from the AJAX response on the screen as a part of the conversation.
5. Add some "intelligence" to the setup:
   a. Display the full conversation history upon page reload.

b. When the bot returns 2 consecutive “not found” answers, return a response with a list of some Star Wars characters using https://swapi.co to get them.
c. When the form message contains the word “force”, print a list of Star Wars films using https://swapi.co to get them.

Once you have completed the challenge, please send an email to
challenge@inbenta.com (Cc. selec.angela@gmail.com),
containing:

- The URL of your page (remember it must be publicly accessible so we can test it)
- The code of the page compressed in a single file
- A state/flow diagram of the new behavior (if you have completed exercise 6)

5. Text Analyzer

The purpose of this test is to evaluate your analysis, design and planning skills. To this end, you need to design a technical solution for the following scenario (don't worry, you won't be coding anything).

We want to create a script to perform different analyses over a set of texts. The script will be defined by the following specifications:

- Texts will be obtained from different configurable sources: databases, APIs, crawler services… Therefore, the script should be flexible enough to add new sources.
- The script must allow to perform one or more analysis in a single execution.
- Currently, we want to perform only one analysis:
  - Finding Lemmas: we want to obtain the number of times a sentence is repeated through all the texts.
- We will want to perform other analysis in the future. Therefore, the script should be flexible enough to add more analysis "algorithms".

We are not asking you to develop all this script (not even how or where it will be executed), but we'd like to see how you would design and plan its development. We don't expect nor want to read more than 2 pages. Also, remember that the reasoning of the solution will be more important than the solution itself. Here are some ideas on what we expect you to answer:

- Short explanation of how the script will be built (UML diagram, explanation of each class and how they are linked…)
- Short explanation on how to manage the configuration of texts sources and which analysis are we going to execute.
- How class dependencies are going to be resolved?
- Have you included some software pattern in the design? Explain why you use them.
- With your proposed solution, how easy will be to include new analysis "algorithms"? And new text sources?
- How complex will be to develop your solution?
- Imagine you have a team of 2 developers full time dedicated to this project. How you will break the development process? Can you give an estimation on how many times will take to develop the solution?

Once you have completed the challenge, please send an email to challenge@inbenta.com (Cc. selec.angela@gmail.com), containing:

- An explanation of the solution
- any necessary attachment you think it's necessary

6. API Access and Credentials

API documentation:

- Setup: https://developers.inbenta.io/chatbot/chatbot-api/api-setup
- Routes: https://developers.inbenta.io/chatbot/chatbot-api/api-routes

YodaBot API credentials:

- API Key: *nyUl7wzXoKtgoHnd2fB0uRrAv0dDyLC+b4Y6xngpJDY=*
- Secret: *eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJwcm9qZWN0IjoieW9kYV9jaGF0Ym90X2VuIn0.anf_eerFhoNq6J8b36_qbD4VqngX79-yyBKWih_eA1-HyaMe2skiJXkRNpyWxpjmpySYWzPGncwvlwz5ZRE7eg*

Authorization process pseudo-code:

- Get the access token:

```
$headers = [
      'x-inbenta-key' => '{{apiKey}}',
      'Content-Type' => 'application/json'
];

$body = [
      'secret' => '{{secret}}'
];

$response = $req->post('https://api.inbenta.io/v1/auth', $headers, $body);
$response = json_decode($response);
$accessToken = $response->accessToken;
$expiration = $response->expiration;
```

- Get the API URL:

```
$headers = [
      'x-inbenta-key' => '{{apiKey}}',
      'Authorization' => 'Bearer '.$accessToken
];
$response = $req->get('https://api.inbenta.io/v1/apis', $headers);
$response = json_decode($response);
$chatbotApiUrl = $response->apis->chatbot;
```