



Chatlaw

Interface

Framework

- Django (2005)
- Flask (2010)

Application minimale
en Flask

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return "Hello world !"

if __name__ == "__main__":
    app.run()
```

Templates

Ajouter du
contenu

```
<!doctype html>
<html>
  <body>

    <h1>Hello {{ name }}!</h1>

  </body>
</html>
```

```
def hello_name(user):
    return render_template('hello.html', name = user)
```

Ajax

Asynchronous
JavaScript
And
XML

```
$.ajax({  
    contentType: 'application/json',  
    type: 'POST',  
    data: JSON.stringify({ message: msg, convID: convID }),  
    beforeSend: function() {$("#loader").show();},  
    success: function (data) {/*...*/},  
    complete:function(data) {$("#loader").hide();}  
});
```

HTML

```
<body>
  <div>
    <div id="logsSup">
      <!-- l'historique des répliques-->
      <div id="logs">
        <p id="bot">Coucou, je suis chatlaw, le
          conseiller juridique en ligne.<br/>Dites-moi
          ce qui vous amène...</p>
      </div>
      <!-- le GIF montrant que le bot n'a pas planté-->
      <div id="loader">


---


      </div>
      <!-- la boîte de saisie et le bouton "envoyer" -->
      <div class="message-box">
        <textarea id="message" type="text" placeholder="Message"></textarea>
        <button id="send">Envoyer</button>
      </div>
    </div>

    <script>


---


  </body>
```

Python

```
def index():
    if request.method == "POST":
        if not request.json or not 'message' in request.json or not request.json.get('message'):
            abort(400)

        convID = request.json['convID']
        msg = request.json['message']

        """recherche de correspondance"""
        from test_question import getBestQuestion, predict
        juriClass = predict(path+'/categorisation/modelIrisLP.mdl', msg)
        bestQuestion = getBestQuestion(msg, juriClass)

        """fonction Boyu pour enrichir msg"""
        question = bold(cut(re.sub('\n', '</br>', bestQuestion['questions']), d
        reponse = bold(cut(re.sub('\n', '</br>', bestQuestion['answers']), dico

        return jsonify({"juriClass": decode_classe[juriClass], "bestQuestion":

return render_template("chatlaw.html", convID=uuid.uuid4()), 200
```