

HUGME HELPERS

# BIBLIOTECAS



REST  
Easy

# 0 CONTRATO

```
@Path("/complain")
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
public interface RaichuClient {

    @POST
    @Path("/answer")
    Response answerComplain(ComplainRequest request, @HeaderParam("Authorization") String token);

    @POST
    @Path("/{complainId}/request-evaluate")
    Response requestEvaluation(@PathParam("complainId") String id, @HeaderParam("Authorization") String token);

    @POST
    @Path("/{complainId}/moderate")
    Response requestModeration(@PathParam("complainId") String id, ModerationRequest request, @HeaderParam("Authorization") String token);
}
```

# A CONEXÃO

---

```
public class ClientConnector {  
  
    public static <T> T createClient(Class<T> clazz, String host) {  
        Client client = new ResteasyClientBuilder()  
            .build();  
        WebTarget webTarget = client.target(host);  
        ResteasyWebTarget resteasyWebTarget = (ResteasyWebTarget) webTarget;  
        return resteasyWebTarget.proxy(clazz);  
    }  
}
```

# IMPLEMENTANDO

CONFIGURAÇÕES DE  
AMBIENTE.

CHAMANDO A API

CONVERSÃO DA  
RESPOSTA PARA  
OBJETO  
CONHECIDO

```
public class RaichuConnector {  
  
    private static String host = HugmeResources.getProperty("raichu.api.host");  
    private static String authToken = HugmeResources.getProperty("raichu.api.token");  
  
    private RaichuClient client = ClientConnector.createClient(RaichuClient.class, host);  
  
    public ComplainResponse answerComplain(ComplainRequest complainRequest) {  
        Response response = client.answerComplain(complainRequest, authToken);  
        ComplainResponse complainResponse = convert(response);  
        return complainResponse;  
    }  
  
    private ComplainResponse convert(Response response) {  
        String responseBody = response.readEntity(String.class);  
        if (response.getStatus() != HttpStatus.SC_NO_CONTENT) {  
            return new Gson().fromJson(responseBody, ComplainResponse.class);  
        }  
        return new ComplainResponse();  
    }  
}
```



```
public class RaichuClientIntegrationTest {

    @Test
    public void test_answer_complain_with_success() {
        given()
            .contentType(ContentType.JSON)
            .header("Authorization", authenticationToken)
        .when()
            .body(createComplainRequest())
            .post(ANSWER_COMPLAIN)
        .then()
            .statusCode(CREATED)
        .and()
            .body("message", equalTo(MESSAGE_REQUEST),
                "type", equalTo(ANSWER),
                "ip", equalTo(IP_REQUEST));
    }
}
```

# MOTIVAÇÃO

"ALWAYS LEAVE THE  
CODE YOU'RE  
EDITING A LITTLE  
BETTER THAN YOU  
FOUND IT"

## **The Boy Scout Rule**

Robert C. Martin

Uncle Bob

