**Exercice : JAX-RS**

On se propose de mettre en place un service web RESTful pour la gestion d’articles publiés sur le web. Le service web fournit plusieurs fonctionnalités.

**Ajouter d’un article**

| **R**  **e**  **q**  **u**  **ê**  **t**  **e**  **R**  **é**  **p**  **o**  **n**  **s**  **e** | **POST** | ws/publication |
| --- | --- | --- |
| **Header** | Content-type : application/xml |
| **Body**  **Header Status** |  |
| **Body** |  |

**Afficher la liste de tous les articles**

| **R**  **e**  **q**  **u**  **ê**  **t**  **e** | **GET** |  |
| --- | --- | --- |
| **R**  **é**  **p**  **o**  **n**  **s**  **e** | **Header** |  |
| **Status** |  |
| **Body** |  |

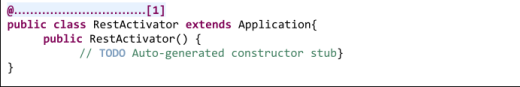
Page **1** sur **5**

|  |  |  |
| --- | --- | --- |

**Travail demandé :**

**1**. **Développez le service web RESTful décrit ci-dessus en complétant les parties manquantes. NB : Veuillez ignorer la logique métier.**

a. Compléter la classe RestActivator



| **[1]** |
| --- |

b. Compléter les classes des entités Article et section

Page **2** sur **5**

2. Soit la classe Publication ci-dessous, Supposant que nous allons ajouter la methode « chercherDetailsArticle » qui permet d’afficher les détails d’un article publié sur le web tel que décrit ci-dessous



| @GET  @Produces("application/xml")  **public** Response chercherDetailsArticle(  @QueryParam(value="ref")String id)  {  …  return Repsonse.status(Status.OK).entity(article).build();  } |
| --- |

Page **3** sur **5**

**Donnez la requête HTTP permattant de consommer le service ainsi que la réponse HTTP retournée en complétant les tableaux ci-dessous avec des exemples concrets.**

| **R**  **e**  **q**  **u**  **ê**  **t**  **e** | **Type de requête** | **GET** |
| --- | --- | --- |
| **URL** | raja3li tableau andou reeference “45” (queryparam) |
| **Header** | - |
| **Body** | **-** |
| **R**  **é** | **Header** | Content-type : application/xml |
| **p**  **o** | **Status** | 200 |

3. Supposant que nous allons ajouter une autre methode qui permet de modifier une section dans un article tel que décrit ci-dessous:

| @PUT  @Path("{ident}")  @Consumes("application/xml")  **public** Response ModifierSection(@PathParam("ident") String id)  {  …  return Repsonse.status(Status.OK).entity(“Article modifié”).build(); } |
| --- |

| **R**  **e**  **q**  **u**  **ê**  **t**  **e** | **Type de requête** |  |
| --- | --- | --- |
| **URL** |  |
| **Header** |  |
| **Body** |  |

Page **4** sur **5**

|  |  |  |
| --- | --- | --- |
| **R**  **é** | **Header** |  |
| **p**  **o** | **Status** |  |

**n**

**s**

**eBody** Article modifié

Page **5** sur **5**