# JS Advanced - Exam: 08.04.2020

## Problem 2. Repository

Use the provided Repository **class** to solve this problem.

### Your Task

Using **Mocha** and **Chai** write **JS Unit Tests** to test the entire functionality of the Repository **class**. Make sure instances of it have all the required functionality and validation. You may use the following code as a template:

|  |
| --- |
| describe(**"*Tests* …"**, **function**() {  describe(**"*TODO* …"**, **function**() {  ***it***(**"*TODO …*"**, **function**() {  *//* ***TODO:*** …  });  });  *//* ***TODO:*** …  }); |

### Functionality

solution.js defines a **class** that contains information about a **Repository class**. An **instance** of the class should support the following operations:

* **Instantiation** with **one** **parameter** - The **props** parameter, which is used to **validate** entities added to the repository and is an object**,** and an **additional property** called data (Map that holds added entities).
* Getter count – returns the number of stored entities
* Function add(entity) – adds an entity to the data; if successful Store entities in a Map where the key is the ID and the value is the entity and returns the resulting ID. Before an entity is **added** to the repository, it should be **validated** against the props object – it needs to have all of the properties that the props object has, and their values must be of the specified type. If **any** property is **missing**, you should **throw** an **Error** with the message: "**Property {propName} is missing from the entity!**". If the property is present, but is of **incorrect** type, **throw** a **TypeError** with the message "**Property {propertyName} is not of correct type!**"**.**
* Function getId(id) – returns the entity with the given ID
* Function update(id, newEntity) – replaces the entity with the given id with the new entity. If the id does **not** exist in the **data** throw an **Error** with message "**Entity with id: {id} does not exist!**". Validate the **new** entity with the **same** validations and **replace** the old one with the new one.
* Function del(id) – deletes an entity by given id. If the id does **not** exist in the **data** throw an **Error** with message "**Entity with id: {id} does not exist!**".

### Examples

|  |
| --- |
| Sample Code Usage |
| let properties = {      name: "string",      age: "number",      birthday: "object"  };  let repository = new Repository(properties);  let entity = {      name: "Pesho",      age: 22,      birthday: new Date(1998, 0, 7)  };  repository.add(entity);  repository.add(entity);  console.log(repository.getId(0));  console.log(repository.getId(1));  entity = {      name: 'Gosho',      age: 22,      birthday: new Date(1998, 0, 7)  };  repository.update(1, entity);  console.log(repository.getId(1));  repository.del(0);  console.log(repository.count); |
| Corresponding Output |
| **{ name: 'Pesho', age: 22, birthday: 1998-01-06T22:00:00.000Z }**  **{ name: 'Pesho', age: 22, birthday: 1998-01-06T22:00:00.000Z }**  **{ name: 'Gosho', age: 22, birthday: 1998-01-06T22:00:00.000Z }**  **1** |

### Submission

Submit your tests inside a describe() statement, as shown above.