**JS Advanced Exam Retake**

**Problem 3. Pet Adoption Agency**

**Your Task**

Write JavaScript unit tests using Mocha and Chai to test the functionality of an object named **petAdoptionAgency**. You may use the following code as a template:

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

The **petAdoptionAgency** object simulates an animal adoption agency and provides several functions for managing pet adoption:

**isPetAvailable(pet, availableCount, vaccinated) -**  A function that accepts three parameters: a **string** that represents the type of pet, a **number** that indicates the quantity of available pets and a **boolean** that indicates whether the pets are vaccinated or not.

* If the **availableCount** is less than or equal to 0, it indicates that there are currently no pets of that specific type available for adoption at the agency. In this case, the function returns:

**`** **Sorry, there are no ${pet}(s) available for adoption at the agency.`**

* If the **availableCount** is greater than 0, the pet is available for adoption, and the function **returns** one of two messages based on the vaccinated parameter:
  + If vaccinated is **true**:

**`** **Great! We have ${availableCount} vaccinated ${pet}(s) available for adoption at the agency.`**

* + If vaccinated is **false**:

**`** **Great! We have ${availableCount} ${pet}(s) available for adoption, but they need vaccination.`**

* + There is a need for validation of the input, the pet parameter should be a **string**, the availableCount parameter should be a **number** and **vaccinated** should be a **boolean**. In case of invalid parameters, the function should **throw an error**:

"Invalid input"

**getRecommendedPets(petList, desiredTraits) -** A function that accepts two parameters: an **array of objects** that stores pet objects with properties "name" and "traits", and a **string** representing the desired traits in a pet.

* The function searches for pets in the **petList** that have traits equal to the received **desiredTraits** and returns a list of their names in the following format:

`Recommended pets with the desired traits (${desiredTraits}): ${petName1, petName2, …}`

If no matching pets are found, it **returns**:

**`Sorry, we currently have no recommended pets with the desired traits: ${desiredTraits}.`**

* You need to validate the input; if **petList** and desiredTraits are **not** an **array** and a **string**, the function should **throw an error:**

**"Invalid input"**

**adoptPet(pet, adopterName)** - A function that accepts two parameters, both **strings.**

* The function **returns** a success message in the format:

**`Congratulations, ${adopterName}! You have adopted ${pet} from the agency. Enjoy your time with your new furry friend!`**

* + It includes input validation: if **pet** is not a string or **adopterName** is not a string, it throws an error with the message:

**"Invalid input."**

**JS Code**

To ease you in the process, you are provided with an implementation that meets all of the specification requirements for the **petAdoptionAgency** object:

|  |
| --- |
| petAdoptionAgency.js |
| const petAdoptionAgency = {      isPetAvailable(pet, availableCount, vaccinated) {        if (typeof pet !== "string" || typeof availableCount !== "number" ||typeof vaccinated !== "boolean") {          throw new Error("Invalid input");        }        if (availableCount <= 0) {          return `Sorry, there are no ${pet}(s) available for adoption at the agency.`;        } else {          if(vaccinated){              return `Great! We have ${availableCount} vaccinated ${pet}(s) available for adoption at the agency.`;          }else{              return `Great! We have ${availableCount} ${pet}(s) available for adoption, but they need vaccination.`;          }          }      },      getRecommendedPets(petList, desiredTraits) {        let recommendedPets = [];          if (!Array.isArray(petList) || typeof desiredTraits !== "string") {          throw new Error("Invalid input");        }          petList.forEach((pet) => {          if (pet.traits === desiredTraits) {            recommendedPets.push(pet.name);          }        });          if (recommendedPets.length === 0) {          return `Sorry, we currently have no recommended pets with the desired traits: ${desiredTraits}.`;        } else {          return `Recommended pets with the desired traits (${desiredTraits}): ${recommendedPets.join(", ")}`;        }      },      adoptPet(pet, adopterName) {        if (typeof pet !== "string" || typeof adopterName !== "string") {          throw new Error("Invalid input");        }          return `Congratulations, ${adopterName}! You have adopted ${pet} from the agency. Enjoy your time with your new furry friend!`;      },    }; |

**Submission**

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