Write a detailed explanation of how to set up one-to-one, one-to-many, and many-to-many relationships using spring data.

To set up a one-to-one relationship using spring data you must use clear and different names for each association. Otherwise you will get a JsonMappingException. They will default to the property name or can be customized using the rel attribute by customizing the endpoint. You must create two repositories and then extend them through the CrudRepository interface. Then you need to add a Library instance to work with. You then need to create an instance of what you’re working with. Then you must create the associations. You do that by using the HTTP method PUT and a body containing the URI of the resource to bind the association. If its successful then It will return a status 204. If you want to delete and association you must call a DELETE method.

To set up a one-to-many relationship using spring data you must us the @OneToMany or @ManyToOne annotation. It can have the optional @RestResource annotation to customize the association resource. You must add a new entity that will represent the many relationship. Then you must add that relationship to the library class and then create an additional repository that ties with the new entity. You then must associate it using a put request to the URI. Again, if you want to delete the association you send a DELETE method.

To set up a many-to-many relationship using spring data you must define it using the @ManyToMany annotation that can be added to the @RestResource. You then need to create a new model class. You then need to add the different associations. You then need to create another repository interface for the new association. You then need to create the resources so that you can establish the association. You do this by sending a GET request to view the association. Then you make your associations using the PUT method. After that is done send anther GET request to make sure it worked. You can also delete this association by sending a DELETE method.