Annotating a method with the `@Transasctional` annotation will cause all the database calls within the method to be rolled back if a `RuntimeException` is thrown by any of the methods within the annotated method.

Note that a checked `Exception being thrown` will **NOT** cause database calls to be rolled bacl.

```java
@Transactional
public Park createPark(Park park) {
    Park savedPark = parkDao.createPark(park);
    for (String stateCode : park.getStates()) {
        parkDao.linkParkState(savedPark.getParkId(),
            stateCode);
    }
    park.setParkId(savedPark.getParkId());
    return park;
}
```

If the `parkDao.linkParkState` method throws a RuntimeException, the INSERT in `parkDao.createPark` will be rolled back and a new Park will not be added to the database.

Oh No! My API isn't working!

When you make a request to the server in Postman, what happens? (Check the Status Code)

No Response
The Request Takes a very long time and doesn't show a response or status code → Make Sure Your Server isn't Stuck on a Breakpoint

500
I get back a 500 Status Code (Internal Server Error) → Try visiting the URL in your browser if it is a GET request or use Postman's Visualizer. It likely displays an error page with more details.

501- 599
I get back a 501 – 599 Status Code (Server Error) → That's outside the scope of this diagram, but typically represents a network issue or temporary outage.

Exception
An exception is thrown on the server and Visual Studio pauses as if it was a breakpoint → Read the error message for more information as to what is wrong. You may need to google the error.

200
I get back a 200 Series Response (OK) but the data in the response looks wrong → Put a debugger breakpoint in your controller method and call the method → Does the value your controller method returns have everything it should?
- **No** → Check your Data Access Object code on the server. You may need to do some debugging
- **Yes** → Check the client to make sure that its model has properties matching your server's model. → Make sure all properties have public getters and setters and a parameterless constructor is present

400
I get back a 400 Series Error (Request Error) → Specifically which response code?

400
400 Bad Request → Check your request body to make sure all validation rules are being respected and everything is the right data type. → Check the response body for more details as to what was invalid.

401
401 Unauthorized → You are calling a method with an Authorize attribute or inside of a controller with an Authorize attribute. → Make sure you include a valid (non-expired) bearer token in the Authentication header of your message. → If your JWT has expired, you will need to log in again.

403
403 Forbidden → You are authenticated, but your user doesn't have permission to work with the requested endpoint. → This is a feature, not a bug. Try a user with more permissions.

404
404 Not Found → Your server may be offline, your URL may be incorrect, or you may be requesting details on an item that was deleted. → Make sure your server is running in Visual Studio. → Double check your request path against the server's path.

405
405 Method Not Supported → You are trying to use a REST Verb / Method on a URL that does not support it but supports a different verb. → Double check your URL and verb against the controller's methods.

415
415 Unsupported Media Type → In Postman, make sure your request body is set to Raw and the type is set to JSON. This usually happens when type is left to "Text"

418
418 I'm a Teapot → That's not your app; it's a teapot!

420 / 429
420 / 429 Too Many Requests → You are requesting things too frequently. Wait a few minutes and try again.