Question:

What are the key things you would consider when creating/consuming an API to ensure that it is secure and reliable?

Solution:

While creating an API we should consider the following to make the API secure and reliable:

1. Data validation:

We should use data validation while taking the data from the user which ensure that no wrong data will be added to our backend which can be a security thread also.

1. Exception handling:

There must be a proper exception handling mechanism which ensures that any interruption in the normal flow of execution of the code can be handled gracefully

1. Authorisation and authentication:

This must be implemented which endure that only authorise user can access the endpoint , sensitive data or can perform any action. While making a java supplication all we can apply both authentication and authorisation using spring security framework

1. Encryption:

The API should be protected while transferring the data, in JAVA we can use HTTP protocol and we can also use DTO classes for payloads.

1. Testing of endpoints

ALL the API must be tested to ensure that the results are similar to the expectations.

By considering all those things we can easily secure our API’s and further a documentation must be for the follow up.

**Theoretical Challenge**

Q1. How will you tackle the challenge above?

1. First we need to read the CSV file and make an make an table inside the database matches the columns and represents the values and the formulas as per the CSV file
2. Write a programme which can take the values and process the values according to the given formula by titrating over the cells one by one.
3. Output a new CSV file which will be the product of business logic written in service implementation.

Q2. What type of errors you would you check for?

1. There can be a data validation error like there can be required cell with null value of a cell, cell not present etc. which is supposed to be a Integer as per the formula is not like that, for that we can use various libraries for proper parsing of data.
2. There can be a exception where the normal flow of execution stooped by some reason which can be handled using exception handling.

Q3. How might a user break your code?

1. A user can easily break the code by creating a CSV file with invalid data in it like string in place of Integer, using symbols.
2. A user can make a CSV with a formula which can create a loop results in crashing of applications.
3. A user can create such formula which will take a lots of memory eventually leads to the termination of programme in various cases.

To avoid all these type of issues data validation rules must be there with proper exception handling and the programme must be tested with various input output scenarios.