Context Scenario

Suppose you are both working in two different software development companies, where one provides services to the other. Both companies are located in different countries and work under the agile paradigm, that is, they apply methodologies such as Scrum or extreme programming. Company A just started a project with company B, which is a Web application on which the client-side (front-end) will be developed in Angular 2, and the server-side (back-end) will be developed in Node.js and MongoDB. Back-end and front-end communicate through a REST type service, so this does not present an interface definition as SOAP type services do. This service is being developed by company A almost at the same time as the front-end, which is in charge of company B, therefore, there is no documentation on the resources that the service exposes, nor on the methods available for each of them. Also, company A is the one who has direct contact with the client, so the user stories are provided by this same company, and to whom company, B goes to solve doubts about the functionality. On the other hand, the user stories are being defined and refined as the project progresses, so it is common to find modifications in previously developed elements.

Front-end guide

1. Greet your partner and tell him that you are the one who will develop the front-end, and that as far as you know he is the one with whom you can solve back-end and functionality doubts.
2. Wait for an answer
3. Ask for the IP where the service is published.
4. Wait for a reply
5. Asks how to call the HTTP header where the API Key is sent.
6. Wait for response
7. Confirm that if you use the POSTMAN **(tag - suggested)**
8. Wait for response
9. Ask how to simulate that the wizard is authenticated if the login is not yet ready **(tag)**.
10. Wait for response
11. Answer that no, and that you would have to look for one.
12. Wait for a reply
13. After he responds, ask him for a list of the registered attendees so you can test **(tag)**.
14. Wait for a reply
15. Thank them and ask if the attendee they passed on to you already has appointments set up.
16. Wait for a response
17. Now explain your doubts about the user story 1, what is the name of the REST resource to be consulted?
18. Wait for an answer
19. Comment that in the same story 1 it says that it should be automatically identified when it is a name and when it is a social security number, ask how is the format of that number **(tag - suggested)**.
20. Wait for a response
21. Now ask about what the REST resource for story 2 is called **(tag - suggested)**.
22. Wait for answer
23. Now ask how would you go about finding out the dates when appointments are no longer available.
24. Wait for a response
25. Tell her that you are going to test the data she gave you and then tell her that it worked and that you notice that the dates have a special format. Ask if they have to be sent to the service **(tag)** with the same format.
26. Wait for a response
27. Ask about the story 2, if when saving the validations of the criterion 4 they apply them in the backend.
28. Wait for answer
29. Now from story 3, ask if the REST resource to use is the same as in story 2 **(tag - suggested)**.
30. Wait for response
31. Comment that you are going to do a test and then comment that the service says that the GET method is not enabled. Ask how you would do it there
32. Wait for an answer
33. Comment that you would send the data in JSON, but how would you format it?
34. Wait for answer
35. Now ask on what dates you can find quotes to do a test **(tag - suggested)**.
36. Wait for response

37. Comment that the results come in a nested list in the answer object, and ask what the extra fields in the answer are for.
38. Wait for response
39. Tell him that's all for the moment and thank him for his attention.

Back-end guide

1. Respond to the greeting and tell him/her that he/she can ask you any questions about the service and user stories.
2. Wait for a response
3. Tell him that the IP is 45.65.234.9, and to use port 8080 for development. Also tell him that all requests have to be sent with this API key QWE123RTY345 **(tag)**.
4. Wait for a response
5. He confirms that if it is for the header and that the field must be called key. Also, ask if you are going to use some program to test the requests.
6. Wait for a response
7. Tell him that it is ok and that if he wants he can mail him some saved tests from postman.
8. Wait for a reply
9. Tell him that in addition to the API key he has to send in an encrypted string the wizard id. He also comments that the encryption is done with SHA1 and asks if he has any component to do this encryption **(tag)**.
10. Wait for an answer
11. Tell him that you have one that they used in previous Angular 2 projects, and that you send him a link to download it (invent a link) **(tag - suggested)**.
12. Wait for a response
13. Tell him that so far you only have one assistant (registration number: 890687567) registered, but if he needs another one you can register it.
14. Wait for a response
15. Tell him that he has some appointments registered and that he will have more because they are doing some tests in the backend.
16. Waiting for a reply
17. Responds that it is the patient resource **(tag)**.
18. Wait for response
19. Responds that the social security number is a sequence of 9 numbers only.
20. Wait for response
21. Comment that the resource in story 2 is called appointments **(tag - suggested).**
22. Wait for response
23. Comment that the resource s/he should use to find out the dates when appointments are no longer available is appointments/unavailable days/ and that it is sent to her by query string the attendee's license plate **(tag - suggested)**.
24. Wait for response
25. Comments that the date format that is received in the service is yyyy-mm-ddThh:00
26. Wait for response
27. Comment that the validations are done in the back end **(tag)**.
28. Wait for response
29. Confirms that the resource in story 3 is the same as in story 2.
30. Wait for response
31. Comments that in that resource it was decided to use POST because it is easier to handle in the backend **(tag - suggested).**
32. Wait for response
33. Tell him that the JSON must have this sequence of fields DateIni, DateEnd, name, nss, all in string.
34. Wait for response
35. Comment that there are appointments in the whole month of February this year.
36. Awaiting response
37. Regarding the extra fields, he comments that they are the ones that handle the pagination, and that in the query the page field can be added so that it only returns the elements of the specified page.
38. Wait for answer
39. Tell him that they would be available whenever he likes.