Structure:

1. About yourself, experience (api, serialisers)
2. Nick describes EPAM projects
3. Describe your team (how many persons? What do you response for?)
4. Process, methodologies. What is scrum (-), scrum-masters(-), grooming(-)?
5. How you estimate your tasks?
6. Code review. ( 2 developer, 2 approvals, QA)
7. Linting of code ( rubocop, pretty)
8. Do you use CI? (Gitlab, she doesn’t know what is CI)

### BDs

* Do you have experience with no SQL? (Elastic search) How you do set up search? JSON, rest Clients
* Database normalization
* Joins (left, right, differences) migrations, indexations.
* The most popular algorithm of indexation? ex. Request is too long, what do u do?
* Utilities to see how work requests
* Gem strong\_migration

### Patterns

* SOLID (-)
* DDD
* Query objects, service objects
* Controller contacts with a few models. What do you do? ( I thought he talks about concerns? But mb about services)

### RUBY

* Many threads in ruby. Do you work. With Sideqick? Delay jobs, workers (sos)
* What does it mean “everything is an object in ruby”?
* Proc vs lambdas
* Include module vs extend

### Rails.

* ORM. AR, arel(?)
* Middleware (-is proxy beet win your request and app)
* Web Sockets
* Serialaisers ( JSON API), versionnost

### Algorithms

* Stak vs query
* Data structures
* Memore flow, memory blow (ytechka pamyati)

### Tests

* Mock stub

### Git

* Merge
* Rebase. Interactive rebase

### English

1. What is your favourite development tool?
2. Do you like to travel? Fav country
3. What do you like about ruby?

?

How you became a team leader? Your way?

* Communication (!), overtimes, so so tech skils,

Task:

1. Size of arrays. In C++ and Ruby
2. Return [[1,3,5],[2,4,6]
3. Strebers, donates, games. Which structures will they have?