LESSON

THE API RACE (SYMFONY EDITION)

AUTHOR

SERGIO ELVIRA PEREZ

CODEARTS SOLUTIONS

THE API RACE (SYMFONY EDITION)

The API Race places learners in a backend engineering competition where reliability matters as much as speed. Teams design and test Symfony REST endpoints while responding to changing validation, authentication and error-handling requirements.

Subject and level

Advanced VET backend learners who already understand Symfony controllers and databases and are ready to work with REST, JSON, HTTP status codes and token-based authentication.

Connection to curriculum

The exercise connects API design, data validation, authentication and client-server communication with professional backend development outcomes. It reinforces consistent JSON responses, status codes, secure access and automated or repeatable endpoint testing.

Didactic and pedagogic considerations

A racing metaphor provides urgency while structured pit stops create deliberate moments for refactoring and feedback. Roles such as API architect, tester and integration lead distribute participation and encourage students to evaluate quality beyond simply making an endpoint respond.

Time estimate

3 hours

THE API RACE (SYMFONY EDITION)

Group size

15-20 students

Overall introduction to exercise

Each team starts from an incomplete Symfony API and must deliver a small set of working endpoints. Performance, correctness, validation, documentation and security all contribute to the final result, so students learn that the fastest implementation is not necessarily the best one.

The exercise mirrors an agile backend sprint: requirements evolve, tests expose weaknesses and teams must decide when to refactor. This makes HTTP concepts tangible and develops professional habits around testing, contracts and predictable error handling.

THE API RACE (SYMFONY EDITION)

Exercise as presented to the students

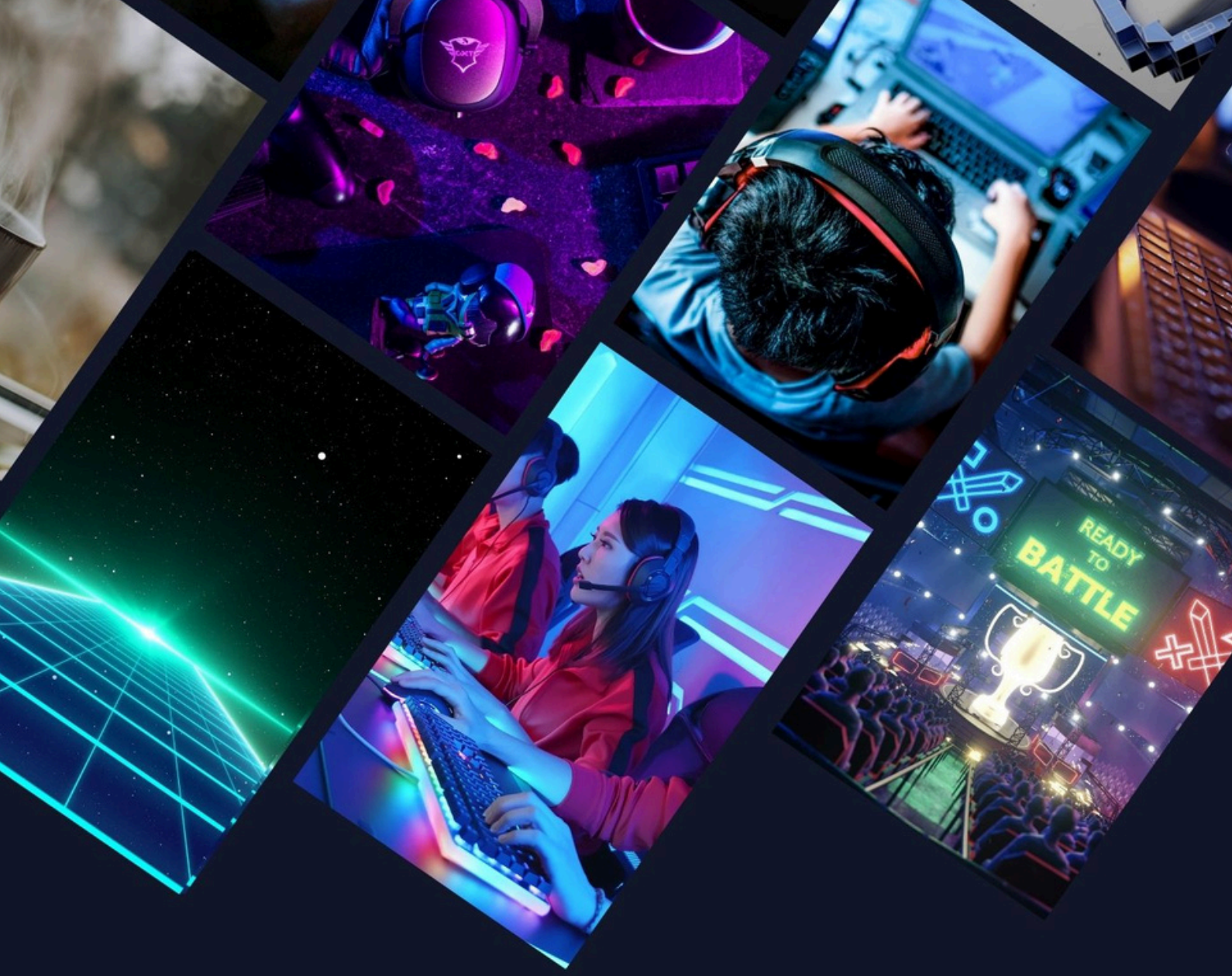
The teacher introduced the challenge as a Formula 1-style API race. Teams became technical pit crews and received the same incomplete Symfony project. Their initial target was clear: deliver endpoints to list records, create a validated record and authenticate a user, all returning consistent JSON and appropriate HTTP status codes.

Students divided responsibilities between implementation, testing and integration. Postman or Insomnia collections were supplied with a few passing and failing examples, but teams had to add their own cases for missing fields, invalid tokens and unknown resources. Each endpoint was checked against a short acceptance checklist before it counted as completed.

At scheduled pit stops the teacher announced new constraints. One round required stricter validation; another introduced a protected endpoint using JWT; a final change required a useful error response instead of a generic server error. Teams paused, reviewed their current design and decided whether a small patch or refactor was safer.

A shared scoreboard displayed completed acceptance criteria rather than raw speed. Bonus points were awarded for readable controllers, reusable services and clear API documentation. This encouraged students to discuss separation of concerns, response consistency and the cost of duplicating logic across endpoints.

The race finished with a short API demo in which another team executed the Postman collection against each solution. The debrief focused on why validation, status codes and authentication form part of the API contract. Students left with a working REST service and practical experience of delivering backend functionality under changing requirements.



EASIE
Esports as Accelerator for Social Inclusive Education

E-SPORT AS ACCELERATOR FOR SOCIAL INCLUSIVE EDUCATION

WWW.EASIE.GG

Co-funded by
the European Union

FUNDED BY THE EUROPEAN UNION. VIEWS AND OPINIONS EXPRESSED ARE HOWEVER THOSE OF THE AUTHOR(S) ONLY AND DO NOT NECESSARILY REFLECT THOSE OF THE EUROPEAN UNION OR THE EUROPEAN EDUCATION AND CULTURE EXECUTIVE AGENCY (EACEA). NEITHER THE EUROPEAN UNION NOR EACEA CAN BE HELD RESPONSIBLE FOR THEM.