

CS 493: Cloud Application Development

Introduction to concepts and techniques for developing and deploying RESTful cloud APIs.

Instructor Eric Ianni

Prerequisites CS 290 and CS 340 and CS 372

Course Content

- RESTful API design and implementation
- Resources, routing, and HTTP verbs
- Data representation and errors
- Efficient data storage models
- Data replication and synchronization
- Authentication and authorization
- Queueing and request processing
- Search indexes and information retrieval
- Cloud API deployment
- API performance, reliability, and scaling

Student Learning Outcomes

At the completion of this course, students will be able to:

- **Design** and implement an effective RESTful API, using appropriate HTTP verbs (GET, POST, DELETE, etc.) to access specific API resources.
- **Evaluate** various approaches to representing data in API requests and responses and to alerting users of errors.
- **Use** modern tools and techniques for storing API data.
- **Use** modern techniques to replicate and synchronize data to ensure data safety and consistency.
- **Employ** secure mechanisms for authenticating users and authorizing the use of specific portions of an API.
- **Use** modern tools and techniques to queue and process API requests.
- **Select** an available open-source search index to meet an API's information retrieval needs.
- **Create** a publicly available cloud API.
- **Evaluate** an API's performance and reliability using appropriate metrics.

Learning Resources

There is no textbook for this course. There will be required articles and other material from the internet posted on the course website as the course progresses.

Evaluation of Student Performance

Final grades will be comprised of the following weighted components:

- 60% Programming assignments
- 35% Final project
 - In lieu of a final exam, you'll demonstrate your mastery of the skills you've learned by the end of this course by working to develop a complete RESTful HTTP API.
- 5% Quizzes or Activities
 - There will be occasional small quizzes or activities which are required

Grades will be assigned using the [standard ranges](#), after rounding. For example:

- $\geq 93\%$ = A
- 90-92% = A-
- 87-89% = B+
- 83-86% = B
- 80-82% = B-
- etc.

(An 89.9 would still be a B+).

Grading for Programming Assignments

For the final project and the assignments it will be your responsibility to provide adequate proof that your API works as intended. This should be something a TA can evaluate quickly. This may require you to record a demo of you interacting with your project via Postman or running a suite of unit tests on your APIs access points.

Generally this will be an easier task for assignments where the TAs are familiar with the API spec. You simply need to demonstrate that it works as intended. For the final project you will need to both demonstrate functionality as well as communicate the overall URL and data structures.

Course Policies

- **Late work.** Assignments submitted after the due date will not be accepted without a documented medical or family emergency and will receive a grade of 0.
- **Contesting grades.** You are allowed to request a regrade on any assignment by contacting the "instructors" in a private question on Piazza. Include what points were taken off and why you feel your project does in fact meet the requirements you were penalized for not meeting.

When requesting a regrade the grade will NEVER be lower than when you made the request. We will not lower a grade for finding something else wrong. That said, please note that while reviewing your request we discover we made a mistake in

taking points but also made a mistake in not deducting points it is possible your grade will remain the same.

- **Communication.** All questions seeking technical help should be posted to Piazza so all students may benefit from the answer. If you have a question that is personal in nature send an email with [493] in the subject.

Expectations for Student Conduct

In this course, you are encouraged to collaborate with your fellow classmates to discuss concepts and high-level approaches to programming assignments. However, you are expected to do your own programming work and may not work with other students to write code for programming assignments. You may never copy the work of another student.

Programming assignment submissions will be checked for similarity against other submissions from the current term and past terms and against work published online. At the instructor's discretion, any student whose work is deemed to be too similar to another person's work will receive a zero for the assignment in question, and the offense will be reported as academic dishonesty to the Office of Student Conduct.

See this page for more details on OSU's Student Conduct Code:
<http://studentlife.oregonstate.edu/code>.

Students with Disabilities

Accommodations for students with disabilities are determined and approved by Disability Access Services (DAS). If you, as a student, believe you are eligible for accommodations but have not obtained approval please contact DAS immediately at 541-737-4098 or at <http://ds.oregonstate.edu>. DAS notifies students and faculty members of approved academic accommodations and coordinates implementation of those accommodations. While not required, students and faculty members are encouraged to discuss details of the implementation of individual accommodations.