

# Chapter Conclusion

In this lesson, we'll look at a quick summary of what we've learned in this chapter.

## WE'LL COVER THE FOLLOWING ^

- Summary
  - Benefits
  - Challenges

## Summary #

- ESIs are a possible implementation of frontend integration and lead to **loose coupling**. The applications are simple web applications that, apart from the ESI tag, have no dependencies on the infrastructure.
- The integration with ESIs has the advantage that the **web pages are completely assembled by the cache and can be displayed directly in the browser**. Therefore, the page isn't delivered unusable in any way due to fragments that still have to be loaded.
- Using a cache together with ESI has the advantage that **fragments can be cached**.
  - This means that static pages and static parts of dynamic pages can be cached, which improves performance.
  - The pages can even be cached and assembled from a CDN that supports ESI, further improving performance.
- The cache can also be used to achieve a certain degree of **resilience**.
  - If a web server fails, the cache can return the old data. In this way, the web page remains available but could potentially return invalid information.

However, the cache must hold the most up-to-date information for this

- However, the cache must hold the pages a long time in order for this to happen.
- In addition, the cache must also check the availability of the services.

## Benefits #

- Web page is always delivered in entirety
- Resilience via cache
- Higher performance via cache
- No code in the browser

## Challenges #

- Uniform look and feel
- Additional server infrastructure necessary

---

That's it for this chapter! In the next one, we'll discuss asynchronous microservices!