Testmiljø for API-testing i et Enonic-prosjekt

For å sette opp et testmiljø parallelt med utviklingsmiljøet i et Enonic-prosjekt med fokus på API-testing:

**1. Opprett et parallelt testmiljø:**

* **Speil utviklingsmiljøet:** Sørg for at testmiljøet er en nøyaktig kopi av utviklingsmiljøet. Dette inkluderer samme versjon av Enonic, databaser og alle nødvendige integrasjoner.
* **Databehandling:** Bruk realistiske test data som speiler produksjonsdata, men anonymiser sensitiv informasjon. Oppdater test dataene regelmessig for å reflektere de siste scenariene.

**2. Verktøy og automatisering:**

* **API Testverktøy:** Bruk verktøy som Postman, REST Assured eller SoapUI for å automatisere API-testene. Disse verktøyene kan hjelpe deg med å lage, kjøre og validere API-kallene.
* **CI/CD Integrasjon:** Innpass API-testene dine i en CI/CD-pipeline for å sikre at hver endring i koden automatisk blir testet før den blir en del av hovedkodebasen.

**3. Kontinuerlig integrasjon (CI) og kontinuerlig distribusjon (CD):**

* **Automatiserte tester:** Sett opp automatiserte tester for API-er som kjøres ved hver kodeendring. Dette kan inkludere enhets- og integrasjonstester for å sikre at alle API-er fungerer som forventet etter endringer.
* **Parallell testutførelse:** Sørg for at API-testene kan kjøres parallelt med utviklingen, uten å forstyrre utviklingsmiljøet. Dette kan oppnås ved å bruke separate testgrener eller virtuelle miljøer.

**4. Overvåking og logging:**

* **Sanntidsovervåking:** Gjennomfør sanntidsovervåking for å fange opp eventuelle feil eller ytelsesproblemer i API-ene under testing.
* **Logganalyse:** Sett opp logging som gir deg detaljert informasjon om API-kall, inkludert feilkoder, svartider og dataoverføring.

**5. Manuell testing og validering:**

* **Exploratory Testing:** Utfør manuell eksplorativ testing for å avdekke uforutsette feil eller ‘Cases’ som kanskje ikke dekkes av automatiserte tester.
* **Test av brukervennlighet:** Selv om fokuset er på API-er, kan det være nyttig å validere at API-ene oppfyller brukerkravene, spesielt for kritiske funksjoner.

**6. Vedlikehold av testmiljøet:**

* **Regelmessige oppdateringer:** Hold testmiljøet oppdatert med de siste endringene fra utviklingsmiljøet. Dette sikrer at testene alltid kjører under forhold som ligner de faktiske driftsmiljøene.
* **Automatisert opprydding:** Implementer skript for å automatisk rydde opp i testmiljøet, for eksempel ved å tilbakestille databaser eller miljøkonfigurasjoner etter hver testrunde.

Disse trinnene støtter både manuell og automatisert testing av API-er i et Enonic-prosjekt. Dette vil bidra til å sikre høy kvalitet på API-ene før de rulles ut til produksjon.