

گزارش‌گیری و خروجی‌های تست‌ها در `pytest`

برای تولید گزارش‌های تست و مشاهده نتایج به فرمت‌های مختلف، `pytest` گزینه‌های مختلفی را ارائه می‌دهد. این گزارش‌ها می‌توانند به شما کمک کنند تا نتایج تست‌ها را به‌طور دقیق‌تری مشاهده کنید و در صورت نیاز، آن‌ها را در قالب‌های مختلف ذخیره کنید.

1. تولید گزارش‌های HTML با استفاده از گزینه `html--`

با استفاده از پلاگین `pytest-html` می‌توانید گزارش‌های تست خود را به فرمت HTML تولید کنید که به‌راحتی قابل مشاهده و اشتراک‌گذاری باشد. برای استفاده از این گزینه باید ابتدا پلاگین را نصب کنید:

نصب پلاگین `pytest-html`

```
pip install pytest-html
```

استفاده از گزینه `html--` برای تولید گزارش HTML

پس از نصب پلاگین، می‌توانید با استفاده از گزینه `html--` گزارش تست‌ها را به فرمت HTML تولید کنید:

```
pytest -html=report.html
```

این دستور باعث می‌شود که `pytest` پس از اجرای تست‌ها گزارشی به فرمت HTML تولید کرده و در فایل `report.html` ذخیره کند. این گزارش شامل جزئیات کامل در مورد هر تست، وضعیت موفقیت یا شکست، و زمان اجرای هر تست خواهد بود.

2. تولید گزارش XML با استفاده از گزینه `junitxml--`

اگر نیاز دارید که گزارش‌های تست خود را در قالب XML تولید کنید، می‌توانید از گزینه `junitxml--` استفاده کنید. این گزارش می‌تواند برای ادغام با ابزارهای CI/CD مانند Jenkins استفاده شود.

استفاده از گزینه `junitxml--` برای تولید گزارش XML

```
pytest -junitxml=report.xml
```

این دستور باعث می‌شود که `pytest` گزارشی به فرمت XML تولید کرده و در فایل `report.xml` ذخیره کند. این فرمت XML معمولاً برای استفاده در سیستم‌های اتوماسیون مناسب است.

3. استفاده از پلاگین `pytest-cov` برای بررسی پوشش کد (Code Coverage)

پلاگین `pytest-cov` به شما این امکان را می‌دهد که پوشش کد را برای تست‌های خود بررسی کنید. این پلاگین به‌ویژه برای شناسایی قسمت‌هایی از کد که تحت پوشش تست‌ها قرار ندارند، مفید است.

نصب پلاگین `pytest-cov`

برای نصب `pytest-cov`، از دستور زیر استفاده کنید:

```
pip install pytest-cov
```

استفاده از `pytest-cov` برای بررسی پوشش کد

پس از نصب پلاگین، می‌توانید از گزینه `cov--` برای بررسی پوشش کد استفاده کنید:

```
pytest --cov=my_project tests/
```

در اینجا:

- `my_project` به `pytest` می‌گوید که پوشش کد را برای پروژه‌ای با نام `my_project` بررسی کند.
- `/tests` پوشه‌ای است که شامل تست‌ها است.

این دستور گزارشی از پوشش کد را نمایش می‌دهد که نشان می‌دهد چه درصدی از کد در تست‌ها پوشش داده شده است.

تولید گزارش پوشش کد به فرمت HTML

برای تولید گزارش پوشش کد به فرمت HTML، می‌توانید از گزینه `cov-report=html--` استفاده کنید:

```
pytest --cov=my_project --cov-report=html
```

این دستور گزارشی از پوشش کد را در پوشه‌ای به نام `htmlcov` تولید می‌کند که شامل یک فایل HTML است که می‌توانید آن را در مرورگر خود مشاهده کنید.

4. استفاده از پلاگین‌های اضافی برای گزارش‌گیری پیشرفته

`pytest` پلاگین‌های مختلفی برای تولید گزارش‌های پیشرفته و تجزیه و تحلیل نتایج تست‌ها دارد. برخی از پلاگین‌های رایج عبارتند از:

- `pytest-xdist`: برای اجرای تست‌ها به صورت موازی و توزیع شده روی چندین هسته.
 - `pytest-trail`: برای همگام‌سازی نتایج تست با سیستم‌های مدیریت تست مانند TestRail.
 - `pytest-splunk`: برای ارسال گزارش‌های تست به سرویس Splunk.
- می‌توانید پلاگین‌ها را از PyPI جستجو کرده و نصب کنید و از آن‌ها برای تولید گزارش‌ها و تجزیه و تحلیل دقیق‌تر استفاده کنید.

نتیجه‌گیری

گزارش‌گیری و بررسی پوشش کد از موارد حیاتی در فرآیند تست‌نویسی است. با استفاده از ابزارهایی مانند `pytest-cov`، `pytest-junitxml` و `html`، می‌توانید گزارش‌های متنوع و دقیق‌تری از نتایج تست‌های خود دریافت کنید. این گزارش‌ها به شما کمک می‌کنند تا وضعیت کلی تست‌ها و پوشش کد را ارزیابی کنید و به بهبود کیفیت کد کمک نمایند.