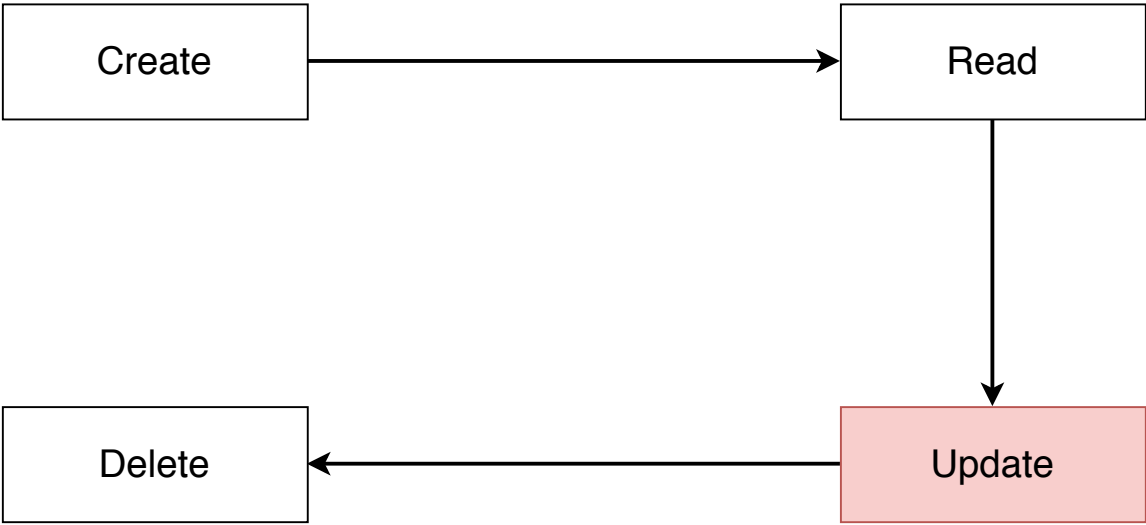
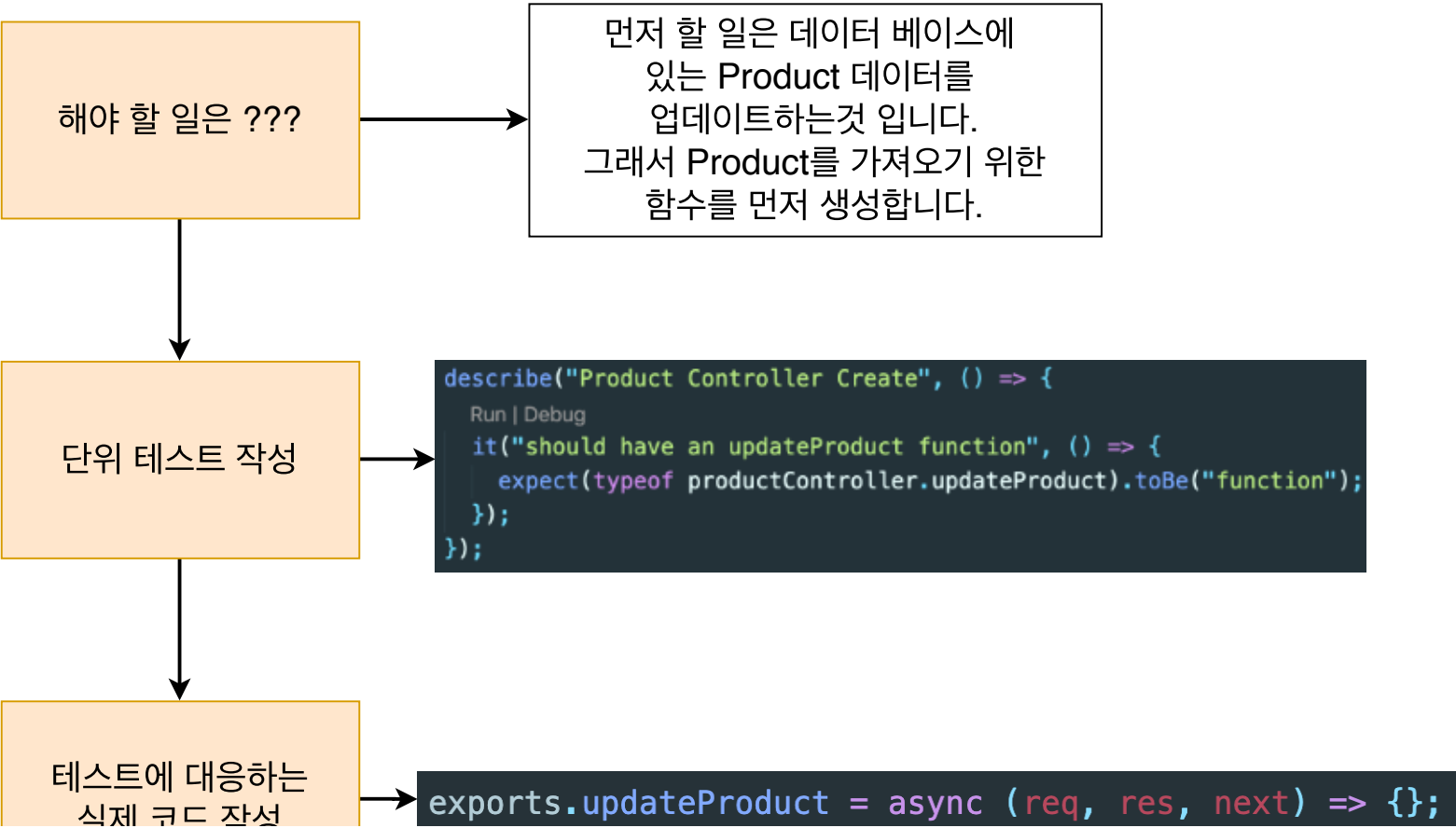


Update 시작

CRUD 작성 순서

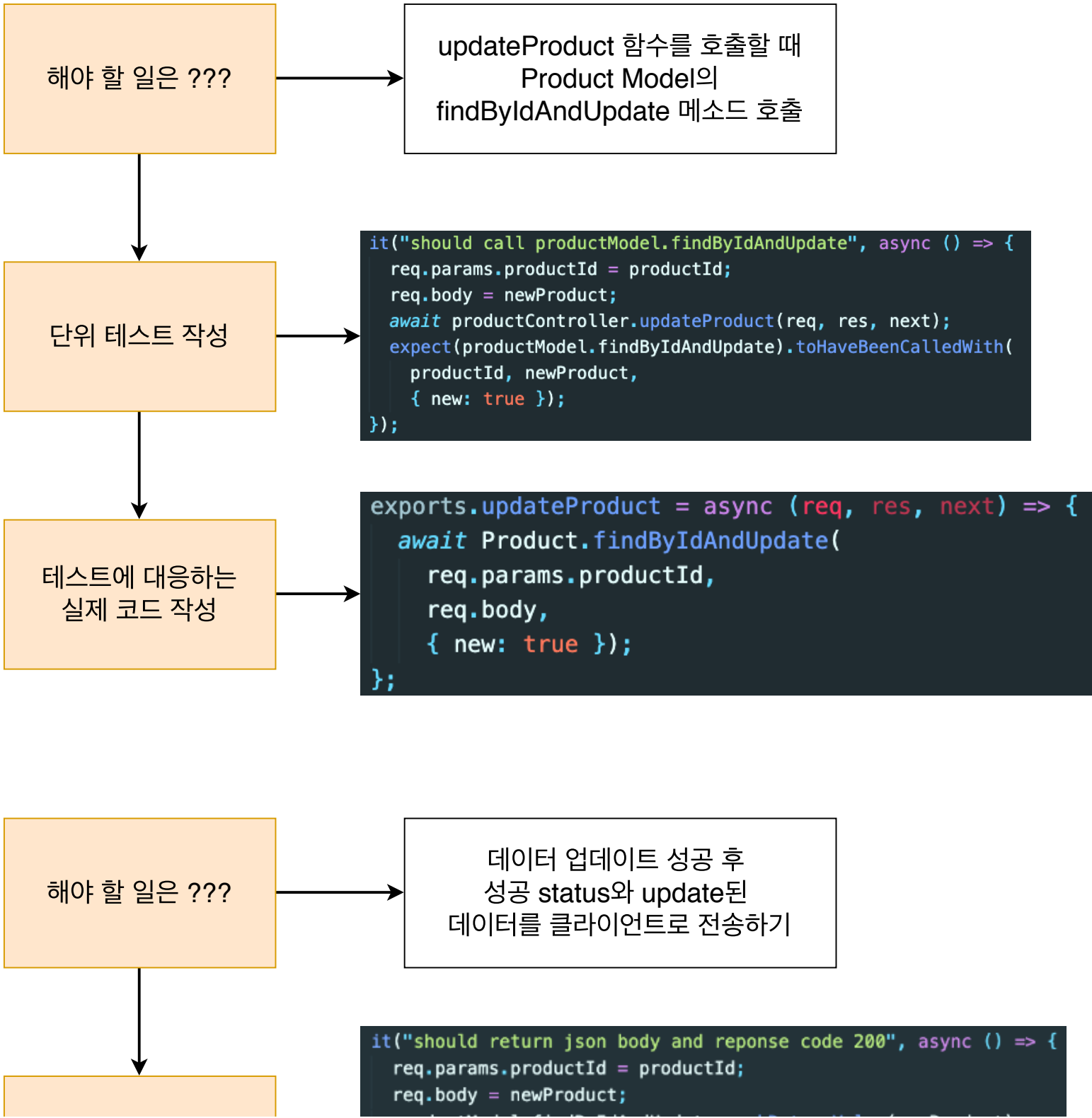


Update 부분 소스 작성



updateProduct 단위 테스트 작성

Update 부분 소스 작성



단위 테스트 작성

```
productModel.findByIdAndUpdate.mockReturnValue(newProduct);  
await productController.updateProduct(req, res, next);  
expect(res._isEndedCalled()).toBeTruthy();  
expect(res.statusCode).toBe(200);  
expect(res._getJSONData()).toStrictEqual(newProduct);  
});
```

테스트에 대응하는
실제 코드 작성

```
exports.updateProduct = async (req, res, next) => {  
  const updatedProduct = await Product.findByIdAndUpdate(  
    req.params.productId,  
    req.body,  
    {  
      new: true  
    }  
  );  
  res.status(200).json(updatedProduct);  
};
```

해야 할 일은 ???

업데이트하는 데이터를
찾지 못한 경우

단위 테스트 작성

```
it("should handle 404 when item doesnt exist", async () => {  
  productModel.findByIdAndUpdate.mockReturnValue(null);  
  await productController.updateProduct(req, res, next);  
  expect(res.statusCode).toBe(404);  
  expect(res._isEndedCalled()).toBeTruthy();  
});
```

테스트에 대응하는
실제 코드 작성

```
exports.updateProduct = async (req, res, next) => {  
  const updatedProduct = await Product.findByIdAndUpdate(  
    req.params.productId,  
    req.body,  
    {  
      new: true  
    }  
  );  
  if (updatedProduct) {  
    res.status(200).json(updatedProduct);  
  } else {  
    res.status(404).send();  
  }  
};
```

해야 할 일은 ???

데이터를 업데이트할 때 에러가 난 경우

단위 테스트 작성

```
it("should handle errors", async () => {  
  const errorMessage = { message: "Error" };  
  const rejectedPromise = Promise.reject(errorMessage);  
  Product.findByIdAndUpdate.mockReturnValue(rejectedPromise);  
  await productController.updateProduct(req, res, next);  
  expect(next).toHaveBeenCalledTimes(1);  
});
```

테스트에 대응하는
실제 코드 작성

```
exports.updateProduct = async (req, res, next) => {  
  try {  
    const updatedProduct = await Product.findByIdAndUpdate(  
      req.params.productId,  
      req.body,  
      {  
        new: true  
      }  
    );  
    if (updatedProduct) {  
      res.status(200).json(updatedProduct);  
    } else {  
      res.status(404).send();  
    }  
  } catch (err) {  
    next(err);  
  }  
};
```

updateProduct 통합 테스트 작성

Update 부분 소스 작성

해야 할 일은 ???

특정 데이터를 업데이트 하기

통합 테스트 작성

```
it("PUT /api/products", async () => {  
  const res = await request(app)  
    .put("/api/products/" + newProductId)  
    .send({  
      name: "updated name",  
      description: "updated description"  
    });  
  expect(res.statusCode).toBe(200);  
  expect(res.body.name).toBe(testData.name);  
  expect(res.body.description).toBe(testData.description);  
});
```

테스트에 대응하는
실제 코드 작성

```
exports.updateProduct = async (req, res, next) => {  
  try {  
    const updatedProduct = await Product.findByIdAndUpdate(  
      req.params.productId,  
      req.body,  
      {  
        new: true  
      }  
    );  
    if (updatedProduct) {  
      res.status(200).json(updatedProduct);  
    } else {  
      res.status(404).send();  
    }  
  } catch (err) {  
    next(err);  
  }  
}
```

