



```
classDiagram
    class CryptoController {
        - cryptoService: CryptoService
        + getCryptos(Pagination): List<Crypto>
        + getCryptoById(Long): Crypto
        + getHistoricalData(long, Date, Date): List<HistorisationCryptoPrice>
    }
    class UserController {
        - userService: UserService
        + registerUser(UserDto): User
        + getUserById(long): User
        + login(UserDto): UserDtoResponse
    }
    class PortfolioController {
        - portfolioService: PortfolioService
        + getPortfolioByUserId(long): List<PortfolioEntry>
        + addPortfolioEntry(Long, long, float): PortfolioEntry
        + deletePortfolioEntry(long): void
        + getPortfolioValue(long): float
    }
    CryptoController --> CryptoService
    UserController --> UserService
    PortfolioController --> PortfolioService
```

The diagram illustrates the structure of three controllers in a system. Each controller is represented by a class box with a header, a list of attributes, and a list of methods.

- CryptoController:**
 - Attribute: `cryptoService: CryptoService`
 - Methods: `getCryptos(Pagination): List<Crypto>`, `getCryptoById(Long): Crypto`, `getHistoricalData(long, Date, Date): List<HistorisationCryptoPrice>`
- UserController:**
 - Attribute: `userService: UserService`
 - Methods: `registerUser(UserDto): User`, `getUserById(long): User`, `login(UserDto): UserDtoResponse`
- PortfolioController:**
 - Attribute: `portfolioService: PortfolioService`
 - Methods: `getPortfolioByUserId(long): List<PortfolioEntry>`, `addPortfolioEntry(Long, long, float): PortfolioEntry`, `deletePortfolioEntry(long): void`, `getPortfolioValue(long): float`

Each controller has a dependency on its corresponding service, indicated by a dashed line with an open arrow pointing from the controller to the service.