

Main 폴더 및 Main.tsx 파일구조 기술 보고서

1. 폴더 및 파일 구조 개요

`components/Main` 폴더는 자산배분, 마켓 타이밍, 전략 등 주요 도메인별로 컴포넌트가 세분화되어 있습니다. 각 도메인별로 하위 폴더(asset, momentum, reEntry, strategy)로 나누어져 있으며, 유지보수와 확장에 최적화된 구조를 갖추고 있습니다.

1.1. 폴더 구조

- `asset/`: 자산군 관리 관련 컴포넌트 (AssetCheckBox, AssetManagement, AssetSearchBox, AssetType, AssetWeight)
- `momentum/`: 모멘텀 전략 관련 컴포넌트 (MomentumBaseLine, MomentumBaseLinePeriod, MomentumBaseMovingAvg, MomentumBoundaryLine, MomentumBoundaryMovingAvg, MomentumBoundaryPeriod, MomentumEntryWeight, MomentumIndex, MomentumLiquidationWeight)
- `reEntry/`: 재진입 전략 관련 컴포넌트 (ReEntryBuyROC, ReEntryMethod, ReEntryPeriod, ReEntrySellROC)
- `strategy/`: 전략 설정 관련 컴포넌트 (StrategyAllCheck, StrategyAlgorithm, StrategyBRB, StrategyNameInput, StrategyRBP, StrategySeed)
- `SettingPeriod.tsx`: 전략 기간 설정 컴포넌트

2. Main.tsx 파일 구조 및 역할

- `Main.tsx`는 전체 전략 설정 페이지의 최상위 컴포넌트로, 각 도메인별 컴포넌트를 조합하여 UI를 구성합니다.
- 상태관리는 zustand 기반의 store에서 selector/action 패턴으로 가져오며, 저장/초기화 등 주요 액션을 담당합니다.
- 주요 섹션: 전략명/알고리즘/시드/리밸런싱, 자산배분, 마켓 타이밍, 전략 기간 등
- 각 도메인별 컴포넌트는 폴더별로 분리되어 import되어 사용됨

3. 구조적 장점

- **도메인별 책임 분리**: 각 기능별로 폴더/컴포넌트가 분리되어 있어 유지보수와 확장에 용이함
- **컴포넌트 재사용성**: 각 도메인별 컴포넌트는 독립적으로 재사용 가능
- **상태관리 일원화**: zustand store를 통해 selector/action 패턴으로 상태를 일원화하여 관리
- **UI/비즈니스 로직 분리**: UI와 상태/비즈니스 로직이 명확히 분리되어 있음

4. 결론

`components/Main` 폴더와 `Main.tsx`는 대규모 전략/자산배분 프로젝트에서 유지보수, 확장성, 가독성, 재사용성을 모두 고려한 구조로 설계되었습니다. 도메인별 폴더 분리와 zustand 기반 상태관리, 컴포넌트 단위의 UI 조합을 통해 실무적 요구사항을 효과적으로 반영하고 있습니다.