UniswapV2Router02 关键调用链（Add Liquidity / Swap）

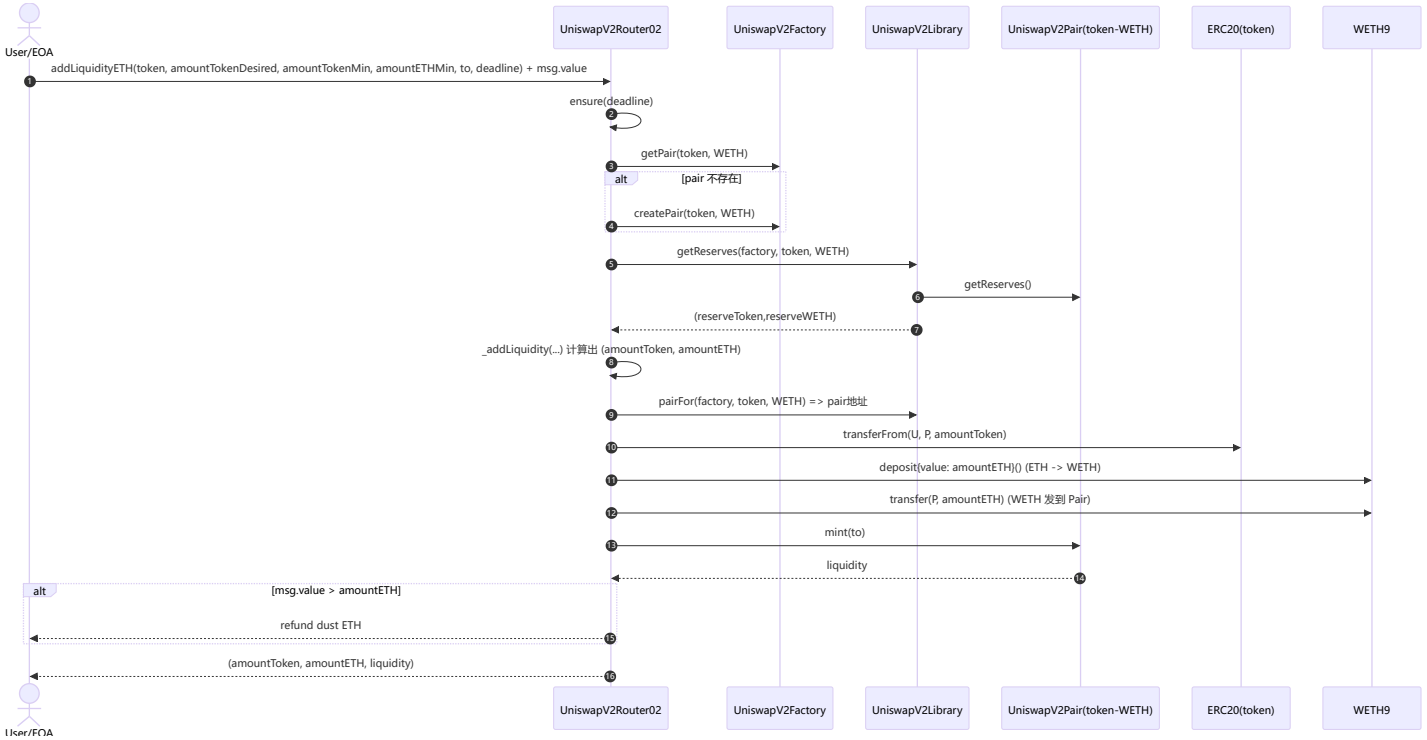
目录

> 说明：本文以本仓库 `src/v2-periphery/contracts/UniswapV2Router02.sol` 的实现为准，重点呈现 Router02 在一次调用中如何串联 `Factory / Pair / WETH / Library`。

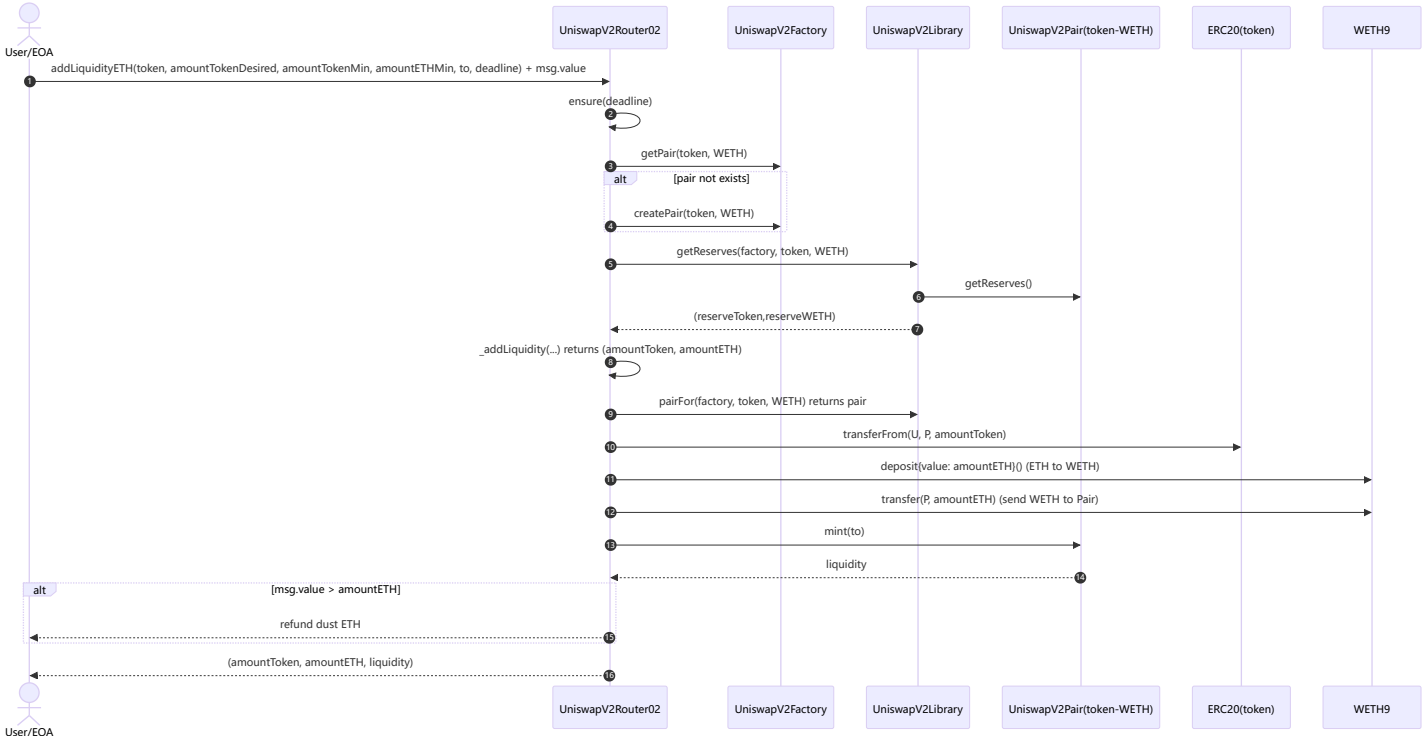
---

1. 加流动性（addLiquidity / addLiquidityETH）

1.1 ERC20-ERC20：addLiquidity(tokenA, tokenB, ...)



1.2 ERC20-ETH：addLiquidityETH(token, ...)



1.3 关键点（加流动性）

- **比例计算在 Router 内**：`_addLiquidity()` 通过 `getReserves()` + `quote()` 决定最终实际转入 Pair 的 `amountA/amountB`。
- **LP 铸造发生在 Pair**：`IUniswapV2Pair(pair).mint(to)`。
- **ETH 路径通过 WETH 进入 Pair**：Router 把 ETH `deposit` 成 WETH 再转入 Pair。

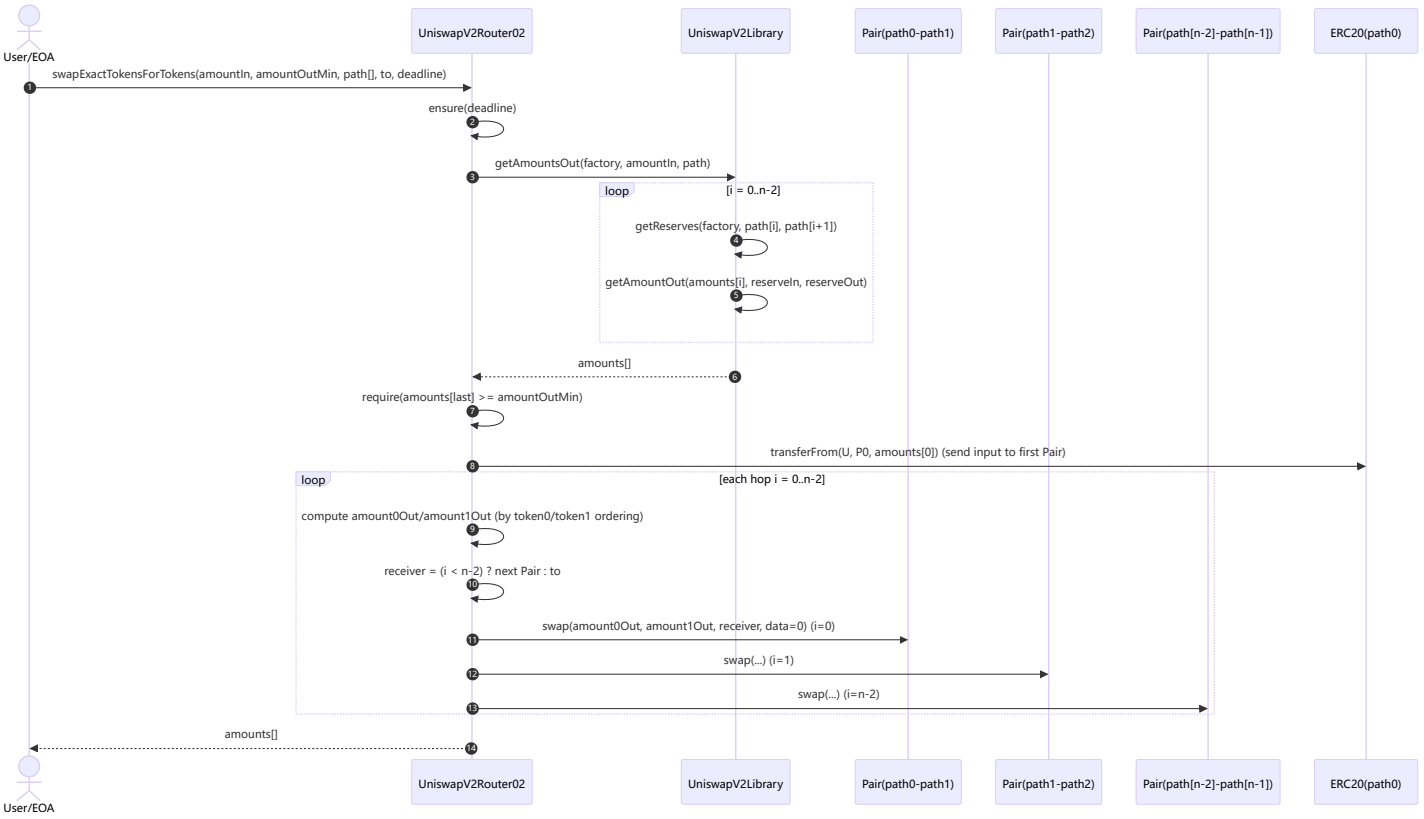
---

2. Swap（普通版 / Fee-on-transfer 版）

2.1 普通版：swapExactTokensForTokens（及其它非 fee-on-transfer 的 swap）

典型结构：

1. Router 调 `UniswapV2Library.getAmountsOut/getAmountsIn` 预计算整条路径 amounts
2. 把 `path[0]` 先转入第一跳 Pair
3. Router `_swap()` 循环每一跳调用 Pair 的 `swap()`，中间输出直接发往下一跳 Pair



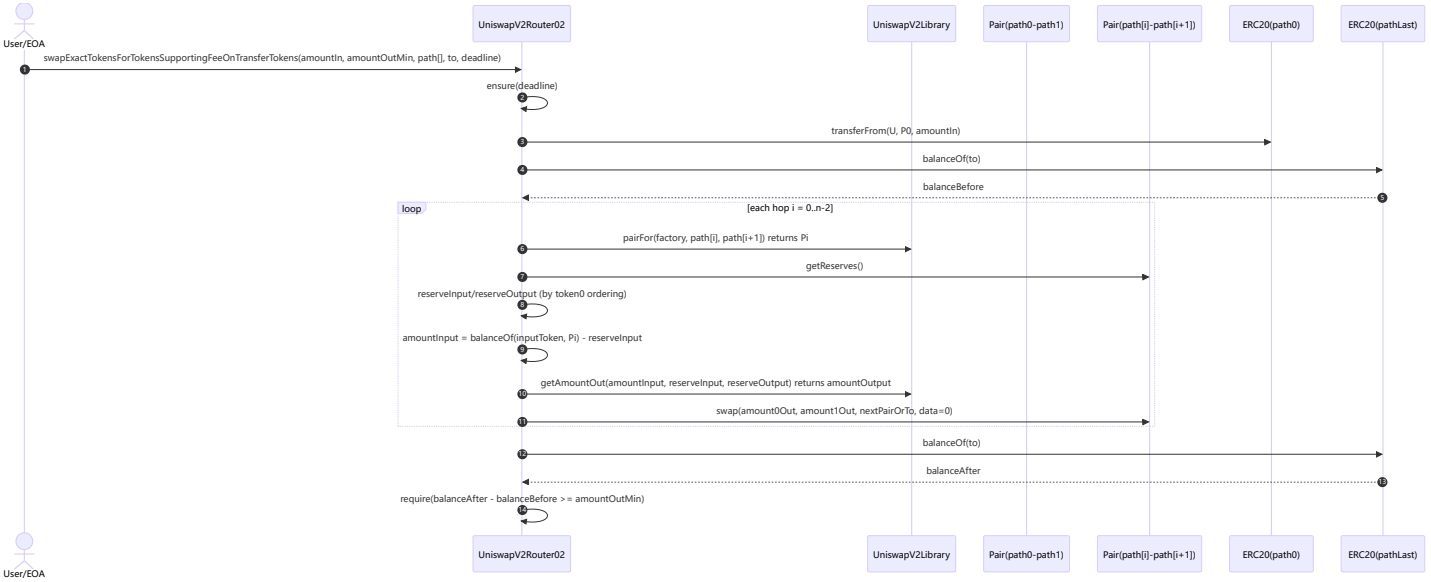
补充（ETH 路径）：

- `swapExactETHForTokens`：先 `WETH.deposit{value: amounts[0]}` 并把 WETH 转给第一跳 Pair，再 `_swap(...)`。
- `swapExactTokensForETH`：先 `_swap(..., address(this))`，最后 `WETH.withdraw(amountOut)` 再转 ETH 给 `to`。

2.2 Fee-on-transfer 版：swapExactTokensForTokensSupportingFeeOnTransferTokens

差异点：

- Router 不再用 `getAmountsOut` 预先精确计算整条 `amounts[]`
- 每一跳的 `amountInput` 用 `balanceOf(pair) - reserveInput` 动态推导（用于适配“转账扣税/反射”等导致实际到帐变化的 token）



2.3 关键点（fee-on-transfer swap）

- **输入以“实际到帐”为准**：`amountInput = IERC20(input).balanceOf(pair) - reserveInput`。
- **最小输出用余额增量校验**：对 `to` 的最终余额做差，确保 `>= amountOutMin`。