Forges' fees

- For **every interest payment** to any user's account, the protocol will take a fees:

`fees = interest calculateed for user * feesFactor`, where 0≤feesFactor<1

- The total fees accrued for the protocol will be kept in a variable `totalFee` , for each of the underlying asset in a forge. I.e. in a mapping `mapping(address => uint256) totalFee`
  - Let's assume that `m(t1, t2)` is how much an amount of yield tokens will grow, from `t1` to `t2`.
    - For Aave: `m(t1, t2)` = NormalisedIncome at t2 / NormalisedIncome at t1
    - For Compound: `m(t1, t2)` = 1
  - This means that if `totalFee` was last updated in `t1` and there is an additional amount of `totalFee` at `t2`:
    - `totalFee = totalFee * m(t1,t2) + additionalFees`
- The governance address could ping the forge at anytime, with a list of underlying asset addresses, to withdraw the `totalFee` to the treasury address