```javascript
import isArray from '../utils/is-array';
import isObject from '../utils/is-object';
import isObjectEmpty from '../utils/is-object-empty';
import isUndefined from '../utils/is-undefined';
import isNumber from '../utils/is-number';
import isDate from '../utils/is-date';
import map from '../utils/map';
import { createInvalid } from './valid';
import { Moment, isMoment } from '../moment/constructor';
import { getLocale } from '../locale/locales';
import { hooks } from '../utils/hooks';
import checkOverflow from './check-overflow';
import { isValid } from './valid';

import { configFromStringAndArray } from './from-string-and-array';
import { configFromStringAndFormat } from './from-string-and-format';
import { configFromString } from './from-string';
import { configFromArray } from './from-array';
import { configFromObject } from './from-object';

function createFromConfig(config) {
    var res = new Moment(checkOverflow(prepareConfig(config)));
    if (res._nextDay) {
        // Adding is smart enough around DST
        res.add(1, 'd');
        res._nextDay = undefined;
    }

    return res;
}

export function prepareConfig(config) {
    var input = config._i,
        format = config._f;

    config._locale = config._locale || getLocale(config._l);

    if (input === null || (format === undefined && input === '')) {
        return createInvalid({ nullInput: true });
    }

    if (typeof input === 'string') {
        config._i = input = config._locale.preparse(input);
    }

    if (isMoment(input)) {
        return new Moment(checkOverflow(input));
    } else if (isDate(input)) {
        config._d = input;
    } else if (isArray(format)) {
        configFromStringAndArray(config);
    } else if (format) {
        configFromStringAndFormat(config);
    } else {
        configFromInput(config);
    }

    if (!isValid(config)) {
        config._d = null;
    }

    return config;
}

function configFromInput(config) {
    var input = config._i;
    if (isUndefined(input)) {
        config._d = new Date(hooks.now());
    } else if (isDate(input)) {
        config._d = new Date(input.valueOf());
    } else if (typeof input === 'string') {
        configFromString(config);
    } else if (isArray(input)) {
        config._a = map(input.slice(0), function (obj) {
            return parseInt(obj, 10);
        });
        configFromArray(config);
    } else if (isObject(input)) {
        configFromObject(config);
    } else if (isNumber(input)) {
        // from milliseconds
        config._d = new Date(input);
    } else {
        hooks.createFromInputFallback(config);
    }
}

export function createLocalOrUTC(input, format, locale, strict, isUTC) {
    var c = {};

    if (format === true || format === false) {
        strict = format;
        format = undefined;
    }

    if (locale === true || locale === false) {
        strict = locale;
        locale = undefined;
    }

    if (
        (isObject(input) && isObjectEmpty(input)) ||
        (isArray(input) && input.length === 0)
    ) {
        input = undefined;
    }
    // object construction must be done this way.
    // https://github.com/moment/moment/issues/1423
    c._isAMomentObject = true;
    c._useUTC = c._isUTC = isUTC;
    c._l = locale;
    c._i = input;
    c._f = format;
    c._strict = strict;

    return createFromConfig(c);
}
```